Frontiers in Artificial Intelligence and Applications

KNOWLEDGE-BASED SOFTWARE ENGINEERING

Proceedings of the Eighth Joint Conference on Knowledge-Based Software Engineering

Edited by
Maria Virvou
Taichi Nakamura

IOS Press

VISIT…

LANZAROTE Caliente.com

KNOWLEDGE-BASED SOFTWARE ENGINEERING

Frontiers in Artificial Intelligence and Applications

Volume 180

Published in the subseries
Knowledge-Based Intelligent Engineering Systems
Editors: L.C. Jain and R.J. Howlett

Recently published in KBIES:

Recently published in FAIA:

ISSN 0922-6389

Knowledge-Based Software Engineering

Proceedings of the Eighth Joint Conference on Knowledge-Based Software Engineering

Edited by

Maria Virvou

Department of Informatics, University of Piraeus, Greece

and

Taichi Nakamura

School of Computer Science, Tokyo University of Technology, Japan

IOS Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

ISBN 978-1-58603-900-4
Library of Congress Control Number: 2008932759

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
Gazelle Books Services Ltd.
White Cross Mills
Hightown
Lancaster LA1 4XS
United Kingdom
fax: +44 1524 63232
e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

PRINTED IN THE NETHERLANDS

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

Preface

This book summarizes the works and new research results presented at the **Eighth Joint Conference on Knowledge-Based Software Engineering 2008 (JCKBSE 2008)**, that took place in August 25–28, 2008 at the University of Piraeus in Piraeus, Greece. JCKBSE is a well established international biannual conference that focuses on applications of artificial intelligence in software engineering. The eighth JCKBSE Conference was organised by the Department of Informatics of the University of Piraeus and it was the first time that the conference took place in Greece. There were submissions from 15 countries.

This year, the majority of submissions originate from Japan (as usual for this conference) and the second place belongs to Greece. The submissions were reviewed rigorously by at least two reviewers per submission and finally 40 papers were accepted as full papers and 16 papers were accepted as short papers. The papers address many topics in the context of Knowledge-Based Software Engineering including new challenges that have arisen in this demanding area of research:

- Knowledge-based requirements engineering, domain analysis and modelling.
- Development processes for knowledge-based applications.
- Knowledge acquisition.
- Software tools assisting the development.
- Architectures for knowledge-based systems and shells including intelligent agents.
- Intelligent user interfaces and human-machine interaction.
- Development of multi-modal interfaces.
- Knowledge technologies for semantic web.
- Internet-based interactive applications.
- Knowledge engineering for process management and project management.
- Methodology and tools for knowledge discovery and data mining.
- Knowledge-based methods and tools for testing, verification and validation, maintenance and evolution.
- Decision support methods for software engineering and cognitive systems.
- Knowledge management for business processes, workflows and enterprise modelling.
- Program understanding, programming knowledge, modeling programs and programmers.
- Software engineering methods for Intelligent Tutoring Systems.

In JCKBSE 2008 we had two important Keynote Speakers, Professor Lakhmi Jain from the University of South Australia, Adelaide, Australia and Professor Xindong Wu from the University of Vermont, U.S.A. who is also Visiting Chair Professor of Data Mining in the Department of Computing at the Hong Kong Polytechnic University, China. Summaries of their talks are included in this book.

In addition there was an invited special session on "Advances in software technologies and cognitive systems" which focused on presenting and discussing such is-

sues from a Research and Development aspect. A summary of the goals of the special session and a paper presented in the special session are included in this book. Finally a tutorial took place on Hybrid Reasoning with Argumentation Schemes.

We would like to thank the authors of the submitted papers for keeping the quality of the conference at high levels. Moreover, we would like to thank the members of the Program Committee as well as the additional reviewers for having performed rigorous reviews of the submissions. For their help with organizational issues of JCKBSE 2008, we express our thanks to the local organizing co-chairs, Professor Dimitris Despotis and Associate Professor George Tsihrintzis, the publicity co-chairs Professor Nikolaos Alexandris and Professor Evangelos Fountas, as well as the local organizing committee members at the University of Piraeus. Thanks are due to Ari Sako of the University of Piraeus for having customized the software Open Conference Manager and for having developed many software modules that facilitated the submission of papers, reviewing process and registration of conference participants.

Maria Virvou and Taichi Nakamura
JCKBSE'08 conference co-chairs

Conference Organization

Conference Co-Chairs:

- Maria Virvou, Department of Informatics, University of Piraeus, Greece
- Taichi Nakamura, School of Computer Science, Tokyo University of Technology, Japan

Steering Committee Members:

- Christo Dichev, Winston-Salem State University, USA
- Pavol Navrat, Slovak University of Technology, Slovakia
- Vadim L. Stefanuk, IITP, Russian Academy of Sciences, Russia
- Shuichiro Yamamoto, NTT Data, Japan
- Seiichi Komiya, Shibaura Institute of Technology, Japan

Program Committee Members:

- Hirohisa Aman, Ehime University, Japan
- Christo Dichev, Winston-Salem State University, USA
- Darina Dicheva, Winston-Salem State University, USA
- Yoshiaki Fukazawa, Waseda University, Japan
- Viktor Gladun, Institute for Cybernetics, Ukraine
- Hele-Mai Haav, Institute of Cybernetics at TUT, Estonia
- Masa-aki Hashimoto, Kyushu Institute of Technology, Japan
- Toyohiko Hirota, Kyushu Sangyo University
- Tadashi Iijima, Keio University, Japan
- Noriaki Izumi, AIST, Japan
- Kenji Kaijiri, Shinshu University, Japan
- Yoshitsugu Kakemoto, The Japan Research Institute
- Shigeo Kaneda, Doshisha University, Japan
- Vladimir Khoroshevski, Computing Centre, Russian Academy of Sciences, Russia
- Mare Koit, University of Tartu, Estonia
- Seiichi Komiya, Shibaura Institute of Technology, Japan
- Mihail Matskin, KTH, Sweden
- Masaru Nakagawa, Wakayama University, Japan
- Takako Nakatani, University of Tsukuba, Japan
- Atsushi Ohnishi, Ritsumeikan University, Japan
- Toshio Okamoto, University of Electro-Communications, Japan
- Gennadii Osipov, Russian Academy of Sciences, Russia
- Themis Panayiotopoulos, University of Piraeus, Greece
- Ivan Rozman, University of Maribor, Slovenia
- Shinobu Saito, NTT Data Corporation, Japan
- Vadim L. Stefanuk, IITP, Russian Academy of Sciences, Russia
- Valery Tarasov, Bauman State Technical Institute, Russia

- Kuldar Taveter, University of Melbourne, Australia
- George Tsihrintzis, University of Piraeus, Greece
- Enn Tyugu, Tallinn University of Technology, Estonia
- Vadim Vagin, Moscow Energy Institute, Russia
- Maria Virvou, University of Piraeus, Greece
- Hironori Washizaki, National Institute of Informatics, Japan
- Rihito Yaegashi, Shibaura Institute of Technology, Japan
- Takahira Yamaguchi, Keio University, Japan
- Shuichiro Yamamoto, NTT Data, Japan

Local Organizing Co-Chairs:
- Dimitris Despotis, University of Piraeus, Greece
- George Tsihrintzis, University of Piraeus, Greece

Publicity Co-Chairs:
- Nikolaos Alexandris, University of Piraeus, Greece
- Evangelos Fountas, University of Piraeus, Greece

Local Organizing Committee Members:
- Efthymios Alepis
- Konstantinos Galanakis
- Chrysostomos Fountas
- Paraskevi Lampropoulou
- Aristomenis Lampropoulos
- Kostantinos Patsakis
- Anastasios Savvopoulos
- Dionysios Sotiropoulos
- Lamprini Stamati
- Ioanna-Ourania Stathopoulou

IT-Technical Support:
Ari Sako

Sponsors:
- Greek Ministry of Education
- University Of Piraeus Research Center
- Department of Informatics, M.Sc. Course in “Informatics”
- Municipality of Piraeus
- Greek Organisation of Tourism

Contents

Knowledge Acquisition

Software Tools Assisting the Development

Architecture of Knowledge-Based Systems or Shells Including Agents

Intelligent User Interfaces and Human-Machine Interaction

Knowledge Engineering for Process Management and Project Management

Methodology and Tools for Knowledge Discovery and Data Mining

Knowledge-Based Methods and Tools for Testing, Verification and Validation, Maintenance and Evolution

Decision Support Methods for Software Engineering

Tutorial on Hybrid Reasoning with Argumentation Schemes

Keynote Addresses

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-3

Knowledge-Based Paradigms and their Applications in Industry

Lakhmi C. Jain,
PhD, ME, BE(Hons), Fellow (Engineers Aust),
Professor of Knowledge-Based Engineering,
Founding Director of the KES Centre,
School of Electrical and Information Engineering,
University of South Australia,
Adelaide,
Mawson Lakes Campus,
South Australia SA 5095,
Australia
Lakhmi.jain@unisa.edu.au

Abstract:

The Knowledge-Based Intelligent Engineering Systems Centre (KES) was established to provide a focus in South Australia for research and teaching activity in the area of intelligent information systems, defence and health industries. The overall goal is to synergise contributions from researchers in disciplines such as engineering, information technology, science, commerce and security.

Knowledge-based paradigms offer many advantages such as learning and generalisation over conventional techniques. Knowledge-based Intelligent Engineering Systems Centre aims to provide applied research support to the Information, Defence and Health Industries.

This talk will report the progress made in some of our research projects undertaken including aircraft landing support, teaming in multi-agent systems and hidden object detection. This talk will mainly focus on the industrial applications of knowledge-based paradigms.

Brief Curriculum Vitae

Professor Lakhmi Jain is a Director/Founder of the Knowledge-Based Intelligent Engineering Systems (KES) Centre, located in the University of South Australia. He is a fellow of the Institution of Engineers Australia. He has initiated a postgraduate stream by research in the Knowledge-based Intelligent Engineering Systems area.

His international standing is evidenced by his appointment to the editorial board of a number of international journals, membership of a number of international conference committees, initiation of an international conference series, a number of invited keynote addresses, initiation of an international journal and position of Co-Editor-in-Chief of a book series by Springer London.

He is the Founding Editor-in-Chief of the International Journal of Knowledge-Based Intelligent Engineering Systems and serves as an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics, Part A.
Dr Jain was the Technical chair of the ETD2000 International Conference in 1995, and Publications Chair of the Australian and New Zealand Conference on Intelligent Information Systems in 1996. He also initiated the First International Conference on Knowledge-based Intelligent Electronic Systems (KES) in 1997. The KES is serving international research community mainly in the area of intelligent systems. The first three KES conferences were held in Adelaide. The KES conference has now attained full international status, with the fourth in the UK, fifth in Japan, the sixth in Italy, the seventh in the UK and the eighth in New Zealand, the ninth in Australia, the tenth in the UK, the eleventh in Italy in 2007. The twelfth KES is scheduled to be held in Croatia in September 2008. This conference is now an established annual international event under the Chairmanship of .Professor Bob Howlett.

He served as the Vice President of the Electronics Association of South Australia in 1997. His interests focus on the applications of novel techniques such as knowledge-based systems, virtual systems, multi-agent intelligent systems, artificial neural networks, genetic algorithms and the application of these techniques. He has worked on many research projects involving complex intelligent paradigms in areas such as robotics, human-machine teaming, learning paradigms, knowledge discovery and intelligent classifiers.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-5

Error-Tolerant Data Mining

Xindong Wu, Professor and Chair of the Department of Computer Science,
University of Vermont,
Burlington, Vermont 05405, USA.
Also Visiting Chair Professor of Data Mining in the Department of Computing
Hong Kong Polytechnic University, China.
Web: http://www.cs.uvm.edu/~xwu/home.html
E-mail: xwu@cs.uvm.edu

Abstract:

Data mining seeks to discover novel and actionable knowledge hidden in the data. As dealing with large, noisy data is a defining characteristic for data mining, where the noise in a data source comes from, whether the noisy items are randomly generated (called random noise) or they comply with some types of generative models (called systematic noise), and how we use these data errors to boost the succeeding mining process and generate better results, are all important and challenging issues that existing data mining algorithms can not yet directly solve.

Consequently, systematic research efforts in bridging the gap between the data errors and the available mining algorithms are needed to provide an accurate understanding of the underlying data and to produce enhanced mining results for imperfect, real-world information sources. This talk presents our recent investigations on bridging the data and knowledge gap in mining noisy information sources.

Brief Curriculum Vitae

Xindong Wu is a Professor and the Chair of the Department of Computer Science at the University of Vermont, USA, and also a Visiting Chair Professor of Data Mining in the Department of Computing at the Hong Kong Polytechnic University, China. He holds a PhD in Artificial Intelligence from the University of Edinburgh, Britain. His research interests include data mining, knowledge-based systems, and Web information exploration. He has published over 160 refereed papers in these areas in various journals and conferences, including IEEE TKDE, TPAMI, ACM TOIS, DMKD, KAIS, IJCAI, AAAI, ICML, KDD, ICDM, and WWW, as well as 18 books and conference proceedings. He won the IEEE ICTAI-2005 Best Paper Award and the IEEE ICDM-2007 Best Theory/Algorithms Paper Runner Up Award.

Dr. Wu is the Editor-in-Chief of the IEEE Transactions on Knowledge and Data Engineering (TKDE, by the IEEE Computer Society), the founder and current Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), the founder and a current Honorary Editor-in-Chief of Knowledge and Information Systems (KAIS, by Springer), the Founding Chair (2002-2006) of the IEEE Computer Society Technical Committee on Intelligent Informatics (TCII), and a Series Editor of the Springer Book Series on Advanced Information and Knowledge Processing (AI&KP). He was Program Committee Chair for ICDM '03 (the 2003 IEEE International Conference on Data Mining) and Program Committee Co-Chair for KDD-07 (the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining).

He is the 2004 ACM SIGKDD Service Award winner, the 2006 IEEE ICDM Outstanding Service Award winner, and a 2005 Chaired Professor in the Cheung Kong (or Yangtze River) Scholars Programme at the Hefei University of Technology sponsored by the Ministry of Education of China and the Li Ka Shing Foundation. He has been an invited/keynote speaker at numerous international conferences including NSF-NGDM'07, PAKDD-07, IEEE EDOC'06, IEEE ICTAI'04, IEEE/WIC/ACM WI'04/IAT'04, SEKE 2002, and PADD-97.

Knowledge-Based Requirements Engineering, Domain Analysis and Modeling

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-9

Classifying Scenarios with Differential Scenario

Atsushi OHNISHI

Department of Computer Science, Ritsumeikan University, JAPAN

Abstract. A visually classification method of scenarios using differential information between normal scenarios is presented. Behaviors of normal scenarios of similar purposes belonging to the same problem domain resemble each other. We derive the differential information, named differential scenario, from such normal scenarios and apply the differential scenario in order to classify scenarios. Our method will be illustrated with examples. This paper describes (1) a language for describing scenarios based on a case grammar of actions, (2) an introduction of the differential scenario, and (3) a method of visualizing scenario classification named Scenario Map using the differential scenario.

Keywords. Scenario classification, Scenario-based requirements elicitation, Scenario analysis

1. Introduction

Scenarios are important in software development [4], particularly in requirements engineering, by providing concrete system description [11, 13]. Especially, scenarios are useful in defining system behaviors by system developers and validating the requirements by customers. In scenario-based software development, incorrect scenarios will have a negative impact on the overall system development process. However scenarios are usually informal and it is difficult to verify the correctness of them. The errors in incorrect scenarios may include (1) vague representations, (2) lack of necessary events, (3) extra events, and (4) wrong sequence among events.

The authors have developed a scenario language named SLAF for describing scenarios in which simple action traces are embellished to include typed frames based on a simple case grammar of actions and for describing the sequence among events [12 14]. Since this language is a controlled language, the vagueness of the scenario written with SLAF language can be reduced. Furthermore, a scenario with SLAF can be transformed into internal representation. In the transformation, both the lack of cases and the illegal usage of noun types can be detected, and concrete words will be assigned to pronouns and omitted indispensable cases [10]. As a result, the scenario with SLAF can avoid the errors typed 1 previously mentioned.

Scenarios can be classified into (1) normal scenarios, (2) alternative scenarios, and (3) exceptional scenarios. A normal one represents the normal behavior of the target system, while an alternative one represents normal but alternative behavior of the system and an exceptional one represents abnormal behavior of the system. In order to grasp whole behaviors of the system, not only normal scenarios, but also alternative/ exceptional scenarios should be specified. However it is difficult to hit upon alternative scenarios and exceptional scenarios, whereas it is easy to think of normal scenarios.

This paper focuses on the classification of scenarios. We adopt SLAF language for writing scenarios, because SLAF is a control language and it is easy to analyze scenarios written with SLAF. Our method will contribute to find similar scenarios and easily get reusable exceptional/alternative scenarios.

2. Scenario Language

2.1 Outline

The SLAF language has already been introduced [14]. In this paper, a brief description of this language will be given for convenience.

A scenario can be regarded as a sequence of events. Events are behaviors employed by users or the system for accomplishing their goals. We assume that each event has just one verb, and that each verb has its own case structure [6]. The scenario language has been developed based on this concept. Verbs and their own case structures depend on problem domains, but the roles of cases are independent of problem domains. The roles include agent, object, recipient, instrument, source, etc. [6, 10].

We provide requirements frames in which verbs and their own case structures are specified. The requirements frame depends on problem domains. Each action has its case structure, and each event can be automatically transformed into internal representation based on the frame. In the transformation, concrete words will be assigned to pronouns and omitted indispensable cases. With Requirements Frame, we can detect both the lack of cases and the illegal usage of noun types [10].

We assume four kinds of time sequences among events: 1) sequence, 2) selection, 3) iteration, and 4) parallelism. Actually most events are sequential events. Our scenario language defines the semantic of verbs with their case structure. For example, data flow verb has source, goal, agent, and instrument cases.

2.2 Scenario example

We consider a scenario of on-line reservation of books at a library system. Figure 1 shows a scenario of customer's retrieving and reserving a book. This scenario is written with our scenario language.

A title of the scenario is given at the first line of the scenario in Figure 1. Viewpoints of the scenario are specified at the second line. In this paper, viewpoints mean active objects such as human and system appearing in the scenario. Active objects

[Title: Scenario1: A customer retrieves and reserves a book on an e-library system]
[Viewpoints: customer, e-library system]
1. A customer enters a title of book to e-library system.
2. The e-library system retrieves books from book-database using the title.
3. The customer gets the result of retrieval from the system.
4. He selects a book for reservation.
5. The system inquires detailed information to the database using ISBN code.
6. The customer gets the detailed information from the system.
7. He enters his user-id and password to the system.
8. **If** (the id and password are correct) **then** he is authenticated by the system.
9. The system reserves the book.
10. The system modifies the database.
11. The customer gets a message of successful reservation from the system.

Figure 1: Scenario example.

are candidates of actors, while data and file are classified into passive objects. There exist two viewpoints, namely customer and system. The order of the specified viewpoints means the priority. In this example, the first prior object is customer, and the second is system. In such a case, the prior object becomes the subject of an event.

In this scenario, almost all events are sequential, except for just one selective event (the 8th event). Selection can be expressed with if-then syntax like program languages. Actually, event number is for reader's convenience and not necessary.

2.3 Analysis of events

Each of events is automatically transformed into internal representation. For example, the first event "A customer enters a title of book to e-library system" can be transformed into internal representation shown in Table 1. In the internal representation, personal pronouns and the definite/indefinite articles are omitted.

Table 1: Internal representation of the 1st event.

Concept: Data Flow

source	**goal**	**object**	**instrument**
customer	e-library system	title of book	NOT specified

In this event, the verb "enter" corresponds to the concept "data flow." The data flow concept has its own case structure with four cases, namely to say, source case, goal case, object case and instrument case. The instrument case is optional, but the others are mandatory. Sender corresponds to the source case and receiver corresponds to the goal case. Data transferred from source case to goal case corresponds to the object case. Device for sending data corresponds to the instrument case. In this event, "title of book" corresponds to the object case and "customer" corresponds to the source case. "E-library system" corresponds to the goal case, but the instrument case object is not specified.

The internal representation is independent of surface representation of the event. Suppose other representations of events, "a title of book is sent from a customer to e-library system" and "e-library system receives a title of book from a customer." These events are syntactically different but semantically same as the first event. These two events can be automatically transformed into the same internal representations as shown in Table 1.

3. Differential Scenario

Systems that are designed for similar purpose (e.g. reservation, shopping, authentication, etc) often have similar behavior. Besides, if such systems belong to the same domain, actors and data resemble each other. In other words, normal scenarios of similar purpose belonging to the same domain resemble each other. Since our scenario language provides limited vocabulary and limited grammar, there exist two benefits, namely, (1) homonym problem can be solved, because limited vocabulary does not permit such problem and (2) the abstraction level of any scenarios becomes almost same, because case structures of actions depend on a certain abstraction level.
For one system, there exist several normal scenarios. In case of ticket reservation, reservation can be written as a normal scenario and cancellation can be written as another normal scenario. To make a differential scenario, we select two normal scenarios of two different systems. Each of the two scenarios should represent almost same purpose, such as reservation of some item.

The differential scenario consists of

1. a list of corresponding words,
2. a list of corresponding events
3. deleted events which appear in one scenario (say, scenario A) and do not appear in the other (say, scenario B), and
4. added events which do not appear in scenario A and appear in scenario B.

We generally assume that one to one correspondence between two nouns and one to one correspondence between two events. Figure 2 shows a scenario of reservation of CD using credit card. We compare the scenario of Fig. 1 (named scenario 1) with the scenario of Fig. 2 (named scenario 2) from the top to the bottom.

First we check the actors specified as viewpoints of the two scenarios. In case of scenarios of Fig. 1 and 2, customer in Fig.1 corresponds to user in Fig.2 and e-library system in Fig.1 corresponds to system in Fig.2. The correspondence should be confirmed by a user.

Second we check action concepts of events. If there exist events whose action concept appears once in scenario 1 and 2, respectively, we assume that these two events are probably corresponding to each other. For example, the concept of the 4th event in Fig.1 and the concept of the 6th event in Fig.2 are "select" and there are no more events whose concepts are "select," we regard these two events are probably corresponding to each other. Then we provide these two events to a user and the user will confirm that these two events are corresponding to each other by checking whether nouns of the same cases are corresponding to or not.

[Title: Scenario 2: A user reserves CD in a rental CD system]
[Viewpoints: User, system]
1. A user enters user-id and password to CD reservation system
2. **If** (user-id and password are correct) **then** the user is authenticated by the system.
3. He enters keywords of CD to the system.
4. The system retrieves CDs from CD-database with the keywords of CD.
5. The user gets candidates from the system.
6. The user selects the candidates.
7. He gives credit card information to the system.
8. The system certifies the credit card information.
9. The system modifies the CD-database
10. The system reserves the CDs.

Figure 2: Normal scenario of CD reservation.

If there are two or more events whose concepts are same in two scenarios respectively, these events are candidates of corresponding events. Then we check whether nouns of the same cases are corresponding to or not. Next we provide candidates to the user and he will select the corresponding event.

The first five events of the scenario in Fig. 1 can be transformed as shown in Table 2. The internal representations of the first five events of the scenario in Fig. 2 are shown in Table 3. In fact, data flow concept has four cases, that is, source, goal, object, and instrument cases as shown in Table 1, but the instrument cases are omitted in Table 2 and 3 for the space limitation.

Table 2: Internal representations of the first 5 events of scenario 1.

concept	Agent/source	goal	object	key
data flow	customer	e-library system	title of book	-
retrieve	e-library system	books	book-database	title
data flow	system	customer	result of retrieval	-
select	customer	NOT specified	book for reservation	-
retrieve	system	detailed information	database	ISBN code

Table 3: Internal representations of the first 5 events of scenario 2.

concept	Agent/source	goal	object	key
data flow	user	CD reservation system	user id and password	-
confirm	system	-	user	-
data flow	user	system	keywords of CD	-
retrieve	system	CDs	CD-database	keywords of CD
data flow	system	user	candidates	-

For the 1st event in Table 2 and the 3rd event in Table 3, these two events are corresponding to each other, because the nouns of the corresponding cases of the two events are same or corresponding to each other. At this time we get "title of book" corresponds to "keywords of CD." Similarly, we detect corresponding events and corresponding words. Finally, we can get a list of corresponding words shown in Table 4.

Table 4: List of corresponding words.

Scenario 1	Scenario 2
customer	user
e-library system	CD reservation system
title of book	keywords of CD
books	CDs
book-database	CD-database
result of retrieval	candidates
book	CDs

We can also get a list of corresponding events, where corresponding events means that the concepts of events are same and nouns assigned to cases are corresponding to each other. Table 5 shows a list of corresponding events between the two scenarios 1 and 2. Some successive events are corresponding to each other, for example first 4 events of the scenario 1 correspond to the 3rd, 4th, 5th and 6th events of the scenario 2. These four events represent a fundamental process of retrieval with keyword.

Table 5: Corresponding events.

Scenario 1	Scenario 2
1. A customer enters …	3. He enters keywords …
2. The e-library system …	4. The system retrieves …
3. The customer gets …	5. The user gets …
4. He selects a book …	6. The user selects …
5. The system inquires …	*No correspondence*
6. The customer gets …	*No correspondence*
7. A customer enters …	1. A user enters user-id …
8. **If** (the id and …	2. **If** (user-id and …
9. The system reserves …	10. The system reserves ...
10. The system modifies ...	9. The system modifies …
11. The customer gets …	*No correspondence*
No correspondence	7. He gives credit …
No correspondence	8. The system certifies …

Since the scenario 1 represents twice retrieval and the scenario 2 represents once retrieval, the behaviors of the second retrieval (5th event and 6th event of scenario 1) are not-corresponding events of the scenario 2. Since the scenario 2 includes payment by credit card, but the scenario 1 does not include such a payment, the 7th event and the 8th events of the scenario 2 are not-corresponding events of the scenario 1. The last event of the scenario 1 is another not-corresponding event of scenario 2. These different points will be stored as added or deleted events of differential scenario. Table 6 shows a list of added events and Table 7 shows a list of deleted events. Finally, we can get the differential scenario between book reservation on an e-library system and CD reservation shown in Table 4, 5, 6, and 7.

Table 6: Added events from perspective scenario1.

Added event	scenario 1	scenario 2
(user) gives (credit card information) to (system)	Payment with cash or no payment	Payment with credit card
(system) certifies (credit card information)	Payment with cash or no payment	Payment with credit card

Table 7: Deleted events from perspective scenario 1.

Deleted event	scenario 1	scenario 2
(System) inquires (detailed information) to (database) using (ISBN code)	Two steps retrieval	One step retrieval
(customer) gets (detailed information) from (system)	Two steps retrieval	One step retrieval
(customer) gets a (successful message of reservation) from (system)	Acceptance message is required	Acceptance message is not required.

4. Scenario Classification using Differential Scenario

In scenario-based software development, several scenarios are specified. Since such scenarios may be revised, there exist a lot of scenarios of different revisions. When a scenario is given, it may be difficult to find similar scenarios or related scenarios to the given scenario by hand. We propose a classification method In order to get similar scenarios or related scenarios using the differential scenario.

We assume that scenarios are analyzed based on the requirements frame in advance. As previously mentioned in 2.1, the requirements frame fully depends on the problem domain. So, if case structures are different between two scenarios, we can regard that these two scenarios are belonging to different domains each other. If case structures are same, these scenarios may be in the same domain. To clarify whether given two scenarios belongs to the same domain, we use the differential scenario.

If differential scenario between two scenarios consists of just corresponding words including human type objects and data type object, we think that these two scenarios represent same behaviors, but systems are different because actors and data are different. For example, suppose two scenarios in Fig. 1 and 2. These scenarios represent purchasing tickets of different transportation. The behaviors of the two systems are similar, but actors and data are different each other. Table 4, 5, 6, and 7 show the differential scenario. There exist seven different but corresponding nouns in Table 4 and in total five different events in Table 6 and 7. In order to exactly clarify the difference of the differential scenario, we introduce the following four measures.

- The ratio of different actors (DA) = The number of different actors / the number of actors in scenario 1
- The ratio of different data (DD) =The number of different data / the number of data in 1
- The ratio of different events (DE) =The number of added and deleted events / the number of events in 1
- Distance between two scenarios (DT) = The number of added and deleted events

In Fig. 1, there exist two actors, that is to say, customer and e-library system and these two actors do not appear in the scenario in Fig. 2. So, DA becomes 1. Similarly, there exist twelve data in Fig. 1 and seven data are different as shown in Table 5, so DD becomes 0.6. DE becomes 0.45, since the number of different events is 5 and the total number of events is 11. These results are shown in Table 8.

Table 8: The 4 differential factors of scenarios of Fig. 1 and 2.

DA	DD	DE	DT
1	0.58	0.45	5

When the differential scenario consists of just added and/or deleted events and almost events of the two scenarios are different, but nouns are same, we think that these two scenarios represent different behaviors of the same system. Suppose two scenarios in Fig. 1 and Figure 3. These scenarios represent slightly different behaviors of the same system each other. In this case, DA becomes 0, because actors of the two scenarios are same. DD becomes 0.25, because the total number of the data of the scenario in Fig. 1 is 12 and three different nouns are added in the scenario in Fig. 3. DE becomes 0.27, because the total number of events of the scenario in Fig. 1 is 11 and three events are added in the scenario in Fig. 3. The four factors of the scenarios are shown in Table 9.

[Title: A customer retrieves books, but cannot find available books, so he gives another title for retrieval.]
[Viewpoints: customer, system]
1. A customer enters a title of book to e-library system.
2. The e-library system retrieves books from book-database using the title.
3. **If** (there are no available books) **then**
 { 3.1. The customer gets a message of no available books from the system.
 3.2. He enters different title to the system.
 3.3. The system retrieves books from the database using the different title.
 }
4. The customer gets the result of retrieval from the system.
5. He selects a book for reservation.
6. The system inquires detailed information to the database using ISBN code.
7. The customer gets the detailed information from the system.
8. He enters his user-id and password to the system.
9. **If** (the id and password are correct) **then** he is authenticated by the system.
10. The system reserves the book.
11. The system modifies the database.
12. The customer gets a message of successful reservation from the system.

Figure 3: An exceptional scenario of book reservation.

Table 10 shows the first three differential factors and relations between two scenarios. In this table "low" means less than 0.5 and "-" means "don't care." The threshold value can be customized through experimental uses and practical uses. When the ratio of different events (DE) is low, we regard that the behaviors of the two scenarios are almost same. If DE becomes high, we regard that the system behaviors of the two scenarios are different each other. In the case of Table 8, we can regard that

these two scenarios of Fig. 1 and 2 represent probably different system of the same domain with almost same behaviors. In the case of Table 9, we can regard that these two scenarios of Fig. 1 and 3 represent probably same system of the same domain with almost same behaviors. By computing these three factors between a certain scenario and other scenarios, we can detect the relation between the scenario and other scenarios.

Table 9: The differential factors of scenarios of Fig. 1 and 3.

DA	DD	DE	DT
0	0.25	0.27	3

Table 10: The first three differential factors and relations between two scenarios.

DA	DD	DE	system	problem domain	behavior
0	0	0	Same	same	same
low	-	low	Probably same	same	almost same
high	-	low	Probably different	same	almost same
low	low	high	Probably same	same	different
low	high	high	Probably different	same	different
high	low	high	Probably different	same	different
high	high	high	Different	different	different

We have developed a prototype system based on the proposed method with C#. We visualize the similarity of scenarios and call it “Scenario Map.” Figure 4 shows the outline of the classification method using the differential scenario and the generation of Scenario Map. Our Scenario Map enables to automatically classify scenarios. With this map, we can select scenarios of similar behaviors and select scenarios of similar systems. In Figure 5, a list of corresponding words and a list of corresponding events are displayed in the right-hand side of the window in generating a differential scenario. A user of the system has to prepare scenarios. At that time, he should specify the corresponding words and corresponding events when the system requires to make differential scenarios. Then he will get a Scenario Map.

Figure 6 shows an example of a Scenario Map. In Fig.6, there exist two axes. The horizontal axis means the similarity of behaviors, while the vertical axis means the similarity of systems. A scenario at the lower left corner is a given scenario. We regard this scenario as a standard scenario. The distance between a given scenario and the standard scenario is measured by DT.

There exist four areas in the Scenario Map. Scenarios placed in the left lower area mean scenarios of same (or similar) system and same (or similar) behavior of the standard scenario. Scenarios placed in the right lower area mean scenarios of same (or similar) system, but different behavior of the standard scenario. Scenarios in the left upper area mean scenarios of same (or similar) behavior, but different system of the standard scenario. Scenarios in the right upper area mean scenarios of different system and different behavior (or different problem domain) of the standard scenario. The distance between a certain scenario and the standard scenario means the degree of similarity between these two scenarios, while the distance between two non standard scenarios does not mean the degree of similarity between them.

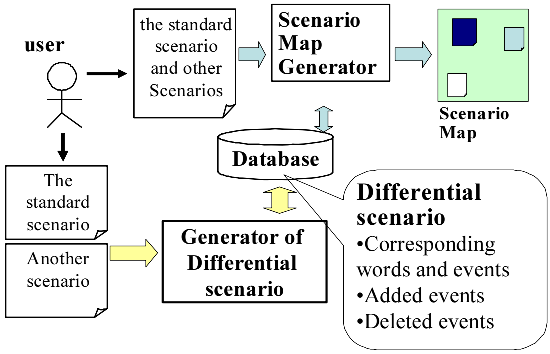


Figure 4: Scenario Map generation with differential scenario.

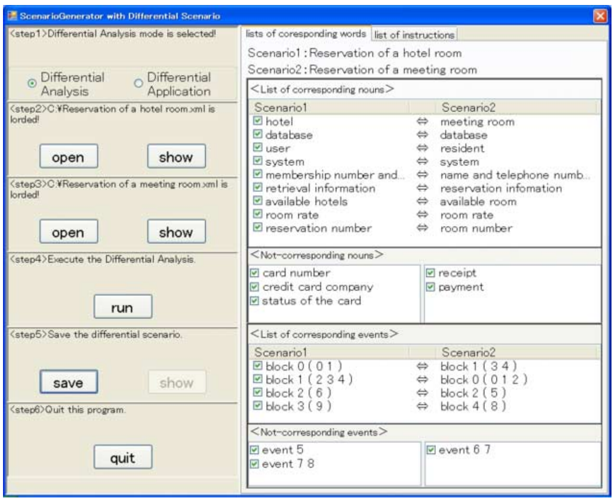


Figure 5: Derivation of the differential scenario.

In Figure 6, the standard scenario represents reservation of flight ticket. The scenario in the left lower area is a revised scenario of the standard scenario. Since the revised scenario is categorized into a scenario of the same system and almost same behavior, it is placed nearby the standard scenario.

Two scenarios in the right lower area specify same system, but different behaviors such as change of flight ticket and cancellation of flight ticket. Three scenarios in the left upper area represent reservation of tickets of other transportation systems, such as train and bus. Two scenarios in the right upper area represent different system behaviors, such as purchase order system, and insurance claim system Titles of these scenarios are also specified in Scenario Map.

To evaluate our method and prototype system, we compare the classification of scenarios by hands with the classification by the system. Thirteen graduate students who well know both the scenario language and the problem domain classify nine scenarios for a standard scenario, while Scenario Map also classifies the same scenarios. The scenario of reservation of flight ticket was adopted as a standard scenario in this experiment. Table 11 shows the comparison of the scenario classifications.

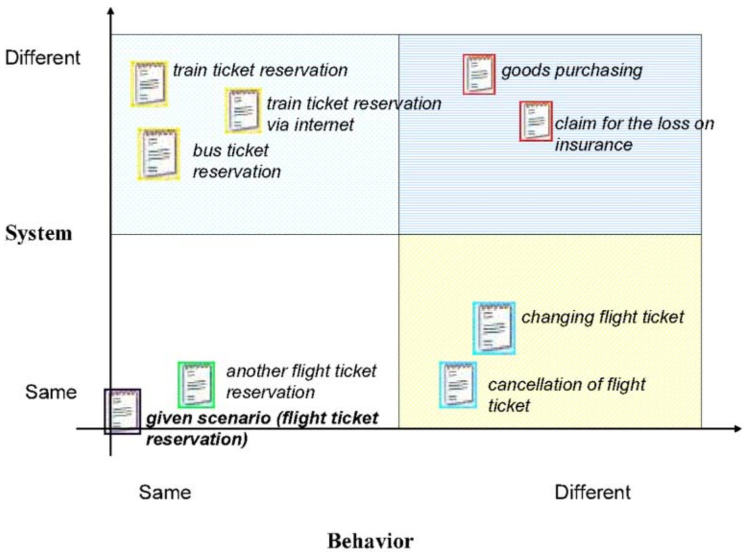


Figure 6: Example of Scenario Map.

In this experiment, nine scenarios are classified. The values of the column "students" mean the ratio of same classification between by Scenario Map and by students. We investigated the reason why some students wrongly classified and found that they did not recognize the difference of systems correctly. Through this experiment, we found that our method and system can correctly classify the scenarios.

Table 11: The comparison of the scenario classification between Scenario Map and students.

scenario	Scenario Map	Same result by students
Train ticket reservation via internet	Different system, same behavior	11/13
Flight ticket changing 1	Same system, different behavior	11/13
Flight ticket changing 2	Same system, different behavior	11/13
Train ticket reservation	Different System, same behavior	11/13
Flight ticket reservation	Same system, same behavior	13/13
Bus ticket reservation	Different System, same behavior	10/13
Claim for the loss on insurance	Different problem domain	13/13
Goods purchasing	Different problem domain	12/13

5. Related Works

There is an obvious trend to define scenarios as textual description of the designed system behaviors. The growing number of practitioners demanding for more "informality" in the requirements engineering process seems to confirm this trend. Most of these papers describe how to use scenarios for the elicitation [11] or exploration [7] of requirements. The authors believe that it is also important to support both the generation and the classification of scenarios

Ben Achour proposed guidance for correcting scenarios, based on a set of rules [1]. These rules aim at the clarification, completion and conceptualization of scenarios, and help the scenario author to improve the scenarios until an acceptable level in terms of the scenario models. Ben Achour's rules can only check whether the scenarios are well written according to the scenario models. We propose generation methods of exceptional scenarios and alternative scenarios from a normal scenario.

Derek Cramp claimed the importance of alternative scenarios. He proposed a model to create alternative scenarios [5]. However, his model strongly depends on a specific domain. Ian Alexander proposed a scenario-driven search method to find more exceptions [2]. In his approach, a model answer was prepared with knowledge of all exception cases identified by stakeholders. For each event, related exceptions are listed as a model answer.

Neil Maiden et al. proposed classes of exceptions for use cases [8]. These classes are generic exceptions, permutations exceptions, permutation options, and problem exceptions. With these classes, alternative courses are generated. For communication actions, 5 problem exceptions are prepared, that is, human agents, machine agents, human-machine interactions, human-human communication, and machine-machine communication. They proposed a generation method of alternative paths for each normal sequence from exception types for events and generic requirements with abnormal patterns [11]. Our approach visualizes the similarity among scenarios using differential scenario. This is a quite new point in the scenario-based requirements engineering researches.

Brian Chance et al. give scenario types for the classification [3]. However they do not propose how to classify given scenarios. Our approach for classifying scenarios gives how to classify given scenarios.

Colette Roland et al. give a classification framework of scenarios in CREWS project [15]. They classify scenarios from four viewpoints. These are contents, purpose, form and lifecycle. They can define the content of a whole scenario. We define contents of a scenario the sequence of actions specified in events and can derive contents of a part of a scenario. In this sense, we can detect similarities between fragments of two different scenarios.

6. Conclusion

We have developed a frame base scenario language and a classification method of scenarios using differential scenario. With our method, the relations among scenarios are easily visualized. We think that our scenario similarity map contributes that system developers and users can easily understand the relations among scenarios.

Actually, the result of scenario classification using Scenario Map in Fig. 6 is almost same as the result of classification by hand. We have to validate the ideas more thoroughly by applying to several different problem domains, such as web-based sales ordering system. We have been developing a prototype system based on the method. The evaluation of our method through the use of the prototype system is another future work.

References

[1] Achour, C. B.: Guiding Scenario Authoring, Proc. of the Eight European-Japanese Conference on Information Modeling and Knowledge Bases, 1998, pp.181-200.
[2] Alexander, I.: Scenario-Driven Search Finds More Exceptions, Proc. 11th International Workshop on Database and Expert Systems Applications, 2000, pp.991-994.
[3] Chance, B.D, Melhart, B.E.: A Taxonomy for Scenario Use in Requirements Elicitation and Analysis of Software Systems, Proc. IEEE Engineering of Computer-Based Systems (ECBS'99), 1999, pp.232-238.
[4] Cockburn, A.: Writing Effective Use Cases, Addison-Wesley, USA, 2001
[5] Cramp, D.G., Carson, E.R.: Assessing Health Policy Strategies: A Model-Based Approach to Decision Support, Proc. International Conference on System, Man and Cybernetics, Vol.3, 1995, pp.69-7.
[6] Fillmore, C.J.: The Case for Case, in Universals in Linguistic Theory, Holt, Rinehart and Winston, 1968.
[7] Leite, J.C.S.P., et al.: Enhancing a Requirements Baseline with Scenarios, Proc. of the 3rd RE, 1997, pp.44-53.
[8] Maiden, N.A.M., et al: CREWS-SAVRE: Systematic Scenarios Generation and Use, Proc. 3rd ICRE'98, 1998, pp.148-155.
[9] Maiden, N.A.M., et al.: Developing Use Cases and Scenarios in the Requirements Process, Proc. 27th ICSE, 2005, pp.561-570.
[10] Ohnishi, A.: Software Requirements Specification Database on Requirements Frame Model, Proc. 2nd ICRE'96, 1996, pp.221-228.
[11] Sutcliffe, A. G., et al.: Supporting Scenario-Based Requirements Engineering, IEEE Trans. Software Engineering, Vol.24, No.12, pp.1072-108, 1998.
[12] Toyama, T., Ohnishi, A.: Rule-based Verification of Scenarios with Pre-conditions and Post-conditions, Proc. 13th IEEE RE2005, 2005, pp.319-328.
[13] Weidenhaupt, K., et al.: Scenarios in System Development: Current Practice, IEEE Software, 1998, pp.34-45.
[14] Zhang, H., Ohnishi, A.: Transformation between Scenarios from Different Viewpoints, IEICE Trans. Information and Systems, Vol.E87-D, No.4, 2004, pp.801-810.
[15] Rolland, C. et al.: A Proposal for a Scenario Classification Framework, CREWS Report 96-01, 1996.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-22

A Conceptual Model for Analysis Method of Extracting Unexpected Obstacles of Embedded Systems

Keiichi KATAMINE [a,1] Yasufumi SHINYASHIKI [b] Toshiro MISE [b]
Masaaki HASHIMOTO [a] Naoyasu UBAYASHI [a] and Takako NAKATANI [c]
[a] *Kyushu Institute of Technology*
[b] *Matsushita Electric Works, Ltd.*
[c] *University of Tsukuba*

Abstract.
This paper proposes a conceptual model for analysis method of extracting unexpected obstacles in order to improve the quality of embedded systems. Although embedded software has become increasingly large in scale and complexity, companies are requiring the software to be developed with in shorter periods of time. This trend in the industry has resulted in the oversight of unexpected obstacles and consequently has affected the quality of embedded software. In order to prevent the oversight of unexpected obstacles, we have already proposed two methods for requirements analysis: the Embedded Systems Improving Method (ESIM) using an Analysis Matrix, and a method that uses an Information Flow Diagram (IFD). However, these analysis methods have been developed separately. This paper proposes the conceptual model including both methods, and clarifies abstraction mechanisms of expert engineers for extracting unexpected obstacles of embedded systems. It also describes a case study and discussion of the domain model.

Keywords. conceptual model, analysis method, formalization, embedded system, unexpected obstacles

1. Introduction

Users of embedded systems are generally unaware of the existence of the systems in different environments [1,2]. Furthermore, these systems are expected to provide safe, reliable service over a long period. As a result, 70% or more of the source code of embedded systems is generally allocated to exception handling. It might be expected that as embedded systems grow larger in scale and become more complicated [3], the development cycle would lengthen. On the contrary, the development cycle has actually shortened, with the result that it has become more difficult to take into account all the exception conditions. Furthermore, software engineers are required to have knowledge not only of the software, but also of the potential devices, users and environments of the software to be able to anticipate exception conditions [4]. The fact that software engineers do occasion-

[1] 680-4 Kawazu, Iizuka, 820-8502 Japan; E-mail: katamine@ci.kyutech.ac.jp

ally overlook exception conditions can only be expected. In fact, the failure of products occurs mainly because exception conditions have not been foreseen, and re-design of the software becomes necessary. This means that the quality and productivity of the embedded software can be improved by reflecting the exception conditions more accurately in the specifications.

We call these exception conditions "unexpected obstacles" and have studied requirements analysis methods to prevent overlooking unexpected obstacles in embedded systems [5,6,7,8,9]. In this paper, the term "expected specifications" specifies the usual system behavior described in the software operation manual and explicitly defined at the start of the architectural design, whereas "unexpected obstacle specifications" are concerned with deviations from the usual behavior. Examples of such deviations are fading and failure of system hardware, incorrect operation or overload caused by system users, and temperature or radio noise in the natural environment. Of course, "unexpected" obstacle specifications should be "expected" and hence explicitly defined in the system specifications. However, they are referred to as "unexpected obstacle specifications" throughout the system development process, in order to distinguish them from expected specifications, and because they are sometimes left undefined.

We have studied two analysis methods for unexpected obstacles by taking account of the procedures that are tacitly used by experts in embedded software development. We have already proposed using an Information Flow Diagram (IFD) in one of the methods [6]. In the other method, which we call ESIM, we proposed the use of an Analysis Matrix [7]. Compared with other existing analysis methods [4,10,11,12,13,14], these methods incorporate both top-down and bottom-up analysis approaches to prevent overlooking unexpected obstacles. Until now, it has been assumed that these two methods can only be used separately. Although the paper [9] proposes an enhanced ESIM which includes IFD in ESIM, it mainly discusses a combination of analysis flows from the viewpoint of an expert engineer or a non-expert engineer. This paper refines the conceptual model in the paper [9], and clarifies abstraction mechanisms of expert engineers for extracting unexpected obstacles. Section 2 clarifies the requirements of the conceptual model and analysis method for extracting unexpected obstacles of embedded systems. Section 3 and 4 describe the conceptual model and analysis method, respectively. Section 5 presents a case study. Section 6 discusses the results.

2. Requirements

This section describes the requirements of the conceptual model and analysis method for extracting unexpected obstacles of embedded systems from the following three viewpoints:

- Concept of embedded systems
  Embedded systems are composed of a variety of kinds of devices such as sensors and controllers. Exception conditions for each device are frequent causes of unexpected obstacle behaviors. Furthermore, embedded systems are operated by users or receive operational triggers from their environment, such as sun light caught by the optical sensors of street-light systems. Exception conditions of the users and external environment are other causes of unexpected obstacle behaviors. Embedded systems identify most causes of unexpected obstacles from hardware; the

embedded software prevents the associated failures by recognizing the hardware information. Therefore, it is necessary to extract not only the devices in the system but also the users and operational environment, and to represent information flow for both hardware and software.

- Analysis from multiple viewpoints
  When the serious exception condition entails a device defect, the assuming the defect phenomenon is not difficult. On the contrary, some exceptions such as deterioration of devices, overload of devices, timing of the operations of users, and environmental influences, are also not important individually. However, the simultaneous occurrence of these phenomena is difficult to assume. Some analysis methods, including top-down analysis like FTA, bottom-up analysis like FMEA, or deflection analysis like HAZOP, are already useful. Even if these techniques are individually applied, multiple occurrences of the exception condition are difficult to find. Therefore, application of these methods is needed to prevent overlooking multiple occurrences of exception conditions.
- Formalization
  In order to analyze multiple occurrences of exception conditions in embedded systems, it is necessary to learn the methods used by expert engineers. However, based on their experience, expert engineers abstract some of the design processes in the interest of design efficiency. Their tacit knowledge has not been formalized, and is useful only for expert engineers themselves. Therefore, the design methods of expert engineers should be analyzed logically, and should be formalized. If a knowledge base developed by formalizing the expert knowledge, it can serve to support the non-expert engineer's analysis.

3. A Conceptual Model

This section proposes a conceptual model to satisfy the requirements described in Section 2. The model represents the structure and constraints of embedded systems. Fig. 1 shows a structure of the conceptual model in a class diagram of UML when expert engineers analyze unexpected obstacles in embedded systems. This model has three parts: "Component" and "Carrier" represent physical components such as hardware devices, and physical communication media between them; "Information Process" and "Information Flow" express an information process and communication model; "State" and "Event" address the state transition model of the embedded systems. Each class has the following meanings:

- Component
  "Component" addresses the basic unit for design. When an expert engineer designs software based on a device, the device is a component. The "Component" represents the hardware, a software component or the operational environment. The component has a four-level hierarchy composed of "Device", "Unit", "Subsystem", and "System" levels. As a result, the component can be represented according to the engineer's level.
- Carrier
  "Carrier" addresses the communication media between components. For example, the electric signals are carriers because the signals carry information between

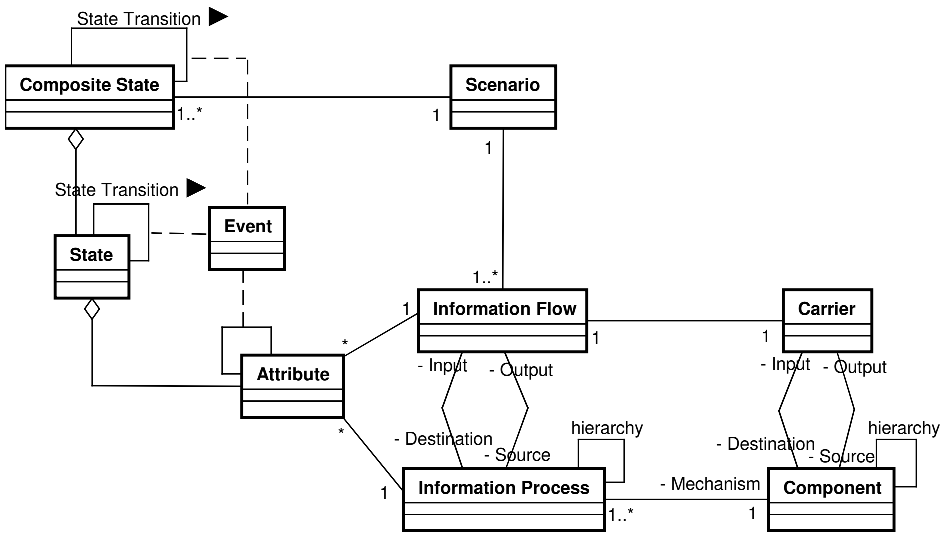


Figure 1. Structure of Conceptual model.

devices. A physical medium such as light or temperature can be a carrier between the operational environment and the embedded systems.

- Information Process
  "Information Process" addresses the information that the component processes. The information process has a four-level hierarchy, just like the components. The information process depends on the component, and uses the component as a mechanism for the process execution.
- Information Flow
  "Information Flow" addresses a flow that connects the information processes with each other. The information flow depends on the carrier.
- Attribute
  "Attribute" means an attribute which has a changeable value. An information process or an information flow has these kinds of changeable attributes. Of course, an attribute value depends on the component or the carrier. This figure omits these links because the link can be represented via an information process or flow.
- State
  "State" addresses a state in a state transition model of an information process or flow. A state is composed of a set of attribute values which the information process or flow has. That is, the state represents a temporary situation of the information process or flow. The state moves to the next state when an event occurs.
- Composite State
  "Composite State" addresses a state in a state transition model of the above-mentioned unit, subsystem or system. The unit, subsystem or system includes some or many components and carriers. Therefore, a composite state is composed of a set of states which the components and carriers have.

- Event
  "Event" addresses an event of the state transition model. Whenever the above-mentioned state changes, an event always happens in this model.
- Scenario
  Scenario "Scenario" means a series of information flows or a sequence of events. The former is constructed with the information process and communication model, and the latter is derived from a state transition model.

The constraints provide the conditions of expected and unexpected specifications for the structure of the conceptual model. There are two kinds of constraints. One is a constraint on attributes of a class on the structure of conceptual model. For instance, the voltage is 5V or less. All classes in the structure of conceptual model have this kind of constraint. Especially, "Component" and "Carrier" classes have failure patterns which enumerate the failures that can occur. The other describes a constraint which should be satisfied through the consequtive information flows. For instance, on a street-light system, an information specifying daytime from the sun, which is the information sender, correctly reaches the software component, which is the information receiver. The constraints are described by using the OCL(Object Constraint Language) [15] of UML.

4. Analysis Method

This section describes the analysis methods by using the conceptual model. First, it explains the three kinds of existing basic methods. Then, it describes the analysis method with IFD and ESIM that uses the above-mentioned methods.

4.1. Existing Basic Analysis methods

Expert engineers analyze unexpected obstacles by using either a top-down analysis like FTA [10], a bottom-up analysis like FMEA [11] or a deflection analysis like HAZOP [12]. When a failure to be prevented is specified in requirement specifications, FTA is used to analyze the cause of the failure specified at the root of a tree. The analysis proceeds from the root to the leaves where the cause is located in a stepwise manner. On contrary, if possible faults of components are known, FMEA is used to analyze the failures that occur as a result of the defaults with the components in a bottom-up manner. HAZOP was originally used for analyzing the safety of plants, and assumes deviational phenomena that have a negative impact on the plant. In a case of software development, engineers assume the deviational phenomena of carriers or information flows. Then, they analyze its causes and resulting failures.

4.2. The Analysis Method with the IFD

We have been developing a method using an IFD, which analyzes unexpected obstacles in embedded systems. The above-mentioned basic analysis methods are applied to this method. The IFD is composed of two diagrams, the Process Diagram (PD) and the Device Diagram (DD), and the connections between these diagrams. The PD represents the processes in the embedded system and the information flow between them by using IDEF0. The DD represents the devices in the embedded system, the objects, such

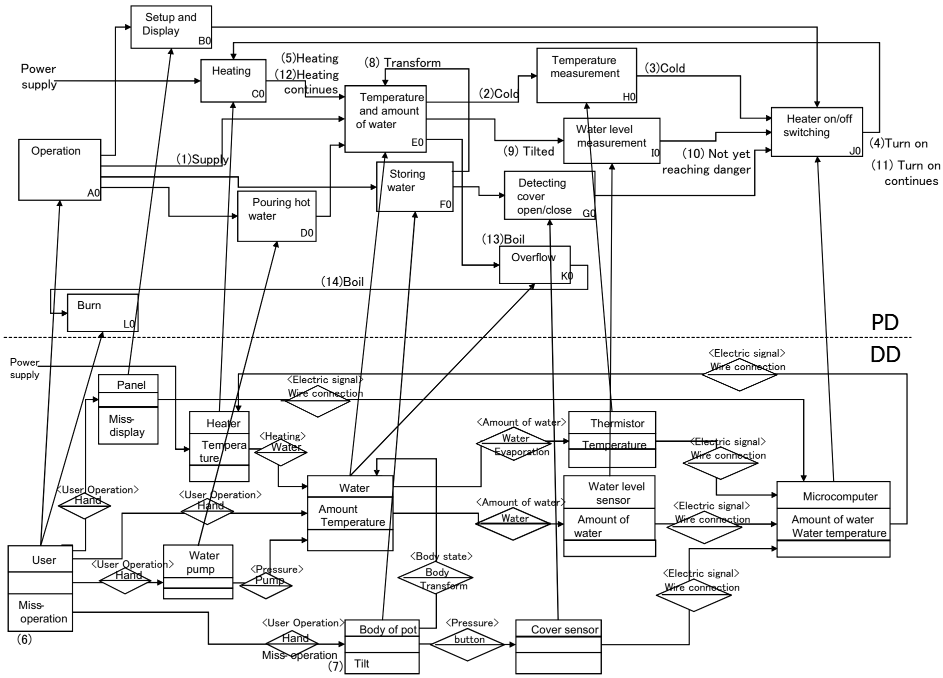


Figure 2. Example of an IFD.

as users, in the operational environment and the communication carriers between them. Fig. 2 shows an example of an IFD. The upper and lower parts are separated by a dotted line, and specify the PD and the DD, respectively.

When we analyze unexpected obstacles with an IFD, we can logically trace the causal relation of the failure from the causes to the final result in the IFD. For example, a failure scenario is outlined by the numbered sequence (1), (2), (3), etc. in the figure. Because of the traceability, non-expert engineers can analyze unexpected obstacles under the leadership of expert engineers. However, much information about devices, carriers, processes and information flows needs to be specified in the IFD.

We discuss the relationship between the conceptual model and the IFD from the viewpoint of the states and events which are important factors in the model. System behaviors are represented by the output of each information process in the IFD. The information process has some attributes which can define the states in the conceptual model. Triggers of state transitions in the IFD are given by the inputs of information processes. Since an information process in the IFD is described by IDEF0, there are three kinds of input information: inputs, controls, and mechanisms. The inputs and the controls correspond to the events sent from other processes, and the mechanisms correspond to the events sent from the components on which the process runs. Values of the attributes of the information process can be changed by events. Therefore, the IFD can be defined in a conceptual model.

4.3. ESIM

We have been developing ESIM to analyze unexpected obstacles and their scenarios from the viewpoint of system behaviors. The ESIM is composed of two phases: a device failure-extracting phase and a failure scenario-constructing phase. In the former phase, by applying guide words, we assume unexpected phenomena to be deviations from the component functions. Then, we extract component device failures from the unexpected phenomena by applying FTA. In the latter phase, we extract unexpected states or events by analyzing unexpected phenomena, and add these states or events to the Analysis Matrix, an example of which is given in Fig. 3. New unexpected phenomena may occur because of the added states or events. This procedure is repeated until no new unexpected phenomena are discovered, and we can then construct the failure scenarios that are often overlooked by the software engineers during the requirements analysis and design phases.

The ESIM has two characteristics. One is that we can find unexpected states and events, by using the ESIM, which are sometimes overlooked by the software engineers. The other is that we can easily obtain all the combinations of states and events on the Analysis Matrix. However, only expert engineers can use the ESIM in practice since the granularity of the information contained in the Analysis Matrix is coarser than that in the IFD. However, the ESIM requires less information to be specified than for the IFD.

The ESIM abstracts some concepts in the conceptual model because the method is based on the thinking processes of expert engineers. We then introduce the following three kinds of abstractions to ensure the consistency of the conceptual model and ESIM.

1. Integration abstraction of serially consequtive associations in the class diagram of the conceptual model
   When expert engineers analyze states and events on the Analysis Matrix, they directly refer the components and carriers. They do not describe the information process and communication model. Therefore, we introduce an abstraction integrating serially consequtive associations from a state to a component or carrier in the class diagram of the conceptual model in oder to ensure consistency between the ESIM and the conceptual model. The information processes and flows on the consecutive associations are abstracted.
2. Integration abstraction of consecutive transitions of states
   Expert engineers pay attention to specific events that lead to the unexpected obstacles in the state transition. This is an abstraction integrating consecutive transitions of states in the state transition. The abstraction is effective for reducing the search domain.
3. Cartesian product abstraction of component states
   ESIM detects an unexpected obstacle within a group of some components. The conceptual model introduces a concept of composite states for representing the cartesian product abstraction of states.

5. Application Example

We show an example of the description of a failure scenario using IFD and ESIM together. The failure scenario is constructed from the sample specifications for an electric

event \ state	The kettle is in an empty state.	The kettle is in the filled of water state.	The water level sensor is in the state of sensing the level.	The kettle is not in the filled of water state and the heater is on.	The water level sensor: sensing the level, but the real water level is out of recognition, The kettle is not in the filled of water state and the heater is on, The kettle is tipped.
Supply water	Turning on the heater	The water may run over.	Reaching the filled of water level	Reaching the filled of water level	Before receiving this event, the kettle may burn the user.
Water level sensor turns off the heater	SAFE	stop boiling water	SAFE	SAFE	SAFE
Something tips the kettle	SAFE	The water may run over.	Out of recognition of the real water level	The water level reaches to the filled water level	------
A user touched the kettle.	SAFE	SAFE	SAFE	SAFE	The kettle may burn the user.

Figure 3. Example of an Analysis Matrix in the ESIM.

kettle. The kettle is equipped with a water sensor that senses the water level. If the sensor detects the filled water state, it turns off the heating element to prevent the boiling water from overflowing. When a user supplies water to the kettle, a thermistor senses the falling temperature, and this event turns on the heating element, which in time causes the kettle to enter the boiling water state. However, there is the possibility of an unexpected event happening, such as something tipping the kettle. Depending on the angle of the kettle, the filled water sensor may not recognize that the actual water level has reached the danger point, and the heating element remains in the state of "boiling". Ultimately, the boiling water overflows and may burn the user. The analysis procedure is as follows:

1. Describe IFD with the expected specifications.
   Fig. 2 except K0 and L0 processes shows it.
2. Describe an Analysis Matrix in ESIM with the expected specifications and known unexpected obstacles.
3. Extract unexpected scenarios by using ESIM.
   In this case when the event that the kettle was tipped happened, two states concerning a water level sensor have been extracted. One state means that a part of water reaches the cap of the kettle though water level is not filled. The other is not the state. The former state is dangerous so that a user may burn himself/herself when the heater keeps on. The expert engineer made the state divided to detect unexpected phenomenon from expert engineers' experience, and he decided each phenomenon. In a word, the first analysis does not care about the state concerning a water level sensor, and expert engineer confirmed the state was made detailed when a specific event happened. Fig. 3 shows final version of the Analysis Matrix.
4. By using IFD, verify the unexpected scenarios extracted by ESIM.
   In Fig. 2, we indicate the failure scenario by attaching numerical comments, e.g. (1), (2), (3), etc.. In Fig. 3, we show the same failure scenario. If we compare the two descriptions, we find that ESIM abstracts the details of the failure scenario. The details corresponding to (2), (3), (5), (7), (8), (9), (11), (12) and (13) in Fig. 2 are not presented in Fig. 3.

6. Discussion

This section discusses the conceptual model and the analysis method, then, describes the future works.

This paper introduced a conceptual model, and fine-tuned a method of analyzing unexpected obstacles by applying the conceptual model. The conceptual model was composed of a structure and constraints. The structure represents the state transition model and the information process and communication model. We confirmed the consistency between ESIM and IFD in the conceptual model.

The constraints can not clearly define the conditions of unexpected obstacles in detail though they can define the conditions of expected specifications. However, it would be possible to support the requirements analysis by enhancing failure patterns. The IFD developed a more detailed with the constraint descriptions.

The conceptual model has made clear the integration abstraction of state transitions in an of ESIM. Moreover, it has became clear that a state transition in the Analysis Matrix includes two kinds of inferences, a logical inference that derives from a different abstraction level, and qualitative reasoning based on the causal relation. Although it is necessary to formalize those two kinds of inferences individually, it is preferable to execute them at the same time from the viewpoint of the design efficiency in practical use. When we formalize them, it is necessary to introduce the abstraction mechanisms and to provide support software design that satisfies both theory and practice at the same time.

The conditions of unexpected obstacles have not been sufficiently clarified, though the constraints of expected specifications are defined in detail. Moreover, operations on the structure of conceptual model in order to formalize analysis procedure in detail are not sufficiently defined, though the structure of IFD and ESIM are defined in the conceptual model. In the future, we will improve on the conceptual model, and define constraints for unexpected obstacles and operations. Moreover, we will systematize expert engineers' knowledge for extracting unexpected obstacles in practical use.

The failure patterns of composite components have not yet been clarified though the patterns for each device have been extracted from case studies. The unexpected obstacles which occur from a composite component are difficult to extract because expert engineers use their tacit knowledge. Therefore, it is necessary to systematize this knowledge and to develop a knowledge base for supporting non-expert engineers' analysis.

7. Conclusion

This paper proposed a conceptual model and described an analysis method by using the model. The conceptual model has a structure and constraints. The structure represents the state transition model and the information process and communication model, and the constraints express conditions of expected specifications. The analysis method requires three basic analysis types: top-down analysis, bottom-up analysis and deflection analysis. IFD and ESIM satisfy these requirements. We have refined the concepts of IFD and ESIM and clarified the abstraction mechanisms based on the conceptual model.

In the future, we will formalize operations for the analysis method. We will also develop a knowledge base by analyzing expert engineers' knowledge for supporting non-expert engineers.

References

[1] T. Nakatani, T. Mise, Y. Shinyashiki, K. Katamine, N. Ubayashi, and M. Hashimoto. Toward defining a system boundary based on real world analysis. In *Proceedings of the Foundation of Software Engineering, FOSE2005*, pages 221–226, November 2005 (in Japanese).

[2] T. Sumi, M. Hirayama, and N. Ubayashi. Analysis of the external environment for embedded systems. In *IPSJ SIG Technical Report*, volume 2004-SE-146(5), pages 33–40, 2003.

[3] Ministry of Economy, Trade and Industry, editor. *Report of actual field survey of embedded software 2005 Edition*. Ministry of Economy, Trade and Industry, 2005 (In Japanese).

[4] R. Crook. Security Requirements Engineering: When Anti-Requirements Hit the Fan. In *Proceedings of Requirement Engineering*, pages 22–33, 2002.

[5] H. Hatanaka, Y. Shinyashiki, T. Mise, H. Kametani, M. Hashimoto, N. Ubayashi, K. Katamine, and T. Nakatani. An Analysis of Information Flow Graph based on Conceptual Model of Exceptions in Embedded Software. In *IEICE Technical Report*, volume 104, pages 19–24. IEICE, November 2004 (in Japanese).

[6] H. Kametani, Y. Shinyashiki, T. Mise, M. Hashimoto, N. Ubayashi, K. Katamine, and T. Nakatani. Information Flow Diagram and Analysis Method for Unexpected Obstacle Specification of Embedded Software. In *The Proceedings of the Knowledge-Based Software Engineering JCKBSE 2006*, pages 115–124, 2006.

[7] T. Mise, Y. Shinyashiki, T. Nakatani, N. Ubayashi, K. Katamine, and M. Hashimoto. A Method for Extracting Unexpected Scenarios of Embedded Systems. In *The Proceedings of the Knowledge-Based Software Engineering JCKBSE 2006*, pages 41–50, 2006.

[8] Y. Shinyashiki, T. Mise, M. Hashimoto, K. Katamine, N. Ubayashi, and T. Nakatani. A Requirements Analysis Method of Unexpected Obstacles in Embedded Software by Applying Information Flow Diagram. *IPSJ Special Issue on Embedded Systems*, 48(9):2894–2903, September 2007 (in Japanese).

[9] Y. Shinyashiki, T. Mise, M. Hashimoto, K. Katamine, N. Ubayashi, and T. Nakatani. Enhancing the ESIM(Embedded Systems Improving Method) by Combining Information Flow Diagram with Analysis Matrix for Efficient Analysis of Unexpected Obstacles in Embedded Software. In *The Proceedings of the 14th ASIA-Pacific Software Engineering Conference*, pages 326–333, December 2007.

[10] N. G. Leveson. Fault Tree Analysis. In *Safeware: System Safety and Computers*, pages 317–326. Addison-Wesley, 1995.

[11] N. G. Leveson. Failure Modes and Effects Analysis. In *Safeware: System Safety and Computers*, pages 341–344. Addison-Wesley, 1995.

[12] N. G. Leveson. Hazards and Operability Analysis. In *Safeware: System Safety and Computers*, pages 335–341. Addison-Wesley, 1995.

[13] I. Alexander. Misuse cases, use cases with hostile intent. In *IEEE Transactions on Software Engineering*, volume 20(1), pages 22–33, 2003.

[14] A. V. Lamsweerde and E. Letier. Handling obstacles in goal-oriented requirements engineering. In *IEEE Transactions on Software Engineering, Special Issue on pp Exception Handling*, volume 26(10), pages 978–1005, 2000.

[15] J. Warmer and A. Kleppe. *The Object Constraint Language – Getting Your Models Ready for MDA, 2nd Edition*. Addison-Wesley, 2004.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-32

A Discussion on Domain Modeling in an Example of Motivation-Based Human Resource Management

Keiichi ISHIBASHI[a,1], Masaaki HASHIMOTO[a], Keiichi KATAMINE[a], Ryoma SHIRATSUCHI[b], Keita ASAINE[b], Takako NAKATANI[c], Naoyasu UBAYASHI[a] and Yoshihiro AKIYAMA[a]
[a]*Kyushu Institute of Technology*
[b]*Juntos Co. Ltd.*
[c]*University of Tsukuba*

Abstract. This paper proposes a domain model comprising a static model, dynamic model and scenario generation method for providing a foundation for motivation-based human resource management. One of the main concerns of managers when establishing a management method in an organization, is the individual members' motivation for the method. It is, however, difficult to manage the members' motivation, because the human resource is very complicated. We therefore, propose a domain model by applying Lawler's motivation model. Using this model, we analyze an actual example of successfully establishing CCPM (Critical Chain Project Management) in a company. We discuss primarily the stability of states, motivation, understanding of individuals' roles and their relationship.

Keywords. Human resource management, Motivation, CCPM (Critical Chain Project Management), O-I (Organization-Individual integration) model, Scenario

1. Introduction

The business environment is changing from day to day. Companies must therefore establish new management methods to adapt to the changing environment. It is however, not easy to switch from an existing management method to another as management is related to the human resource, which is very complicated and very difficult to manage. These issues have been studied using multiple approaches, including human communication, organizational behavior, team building, evaluation of the competency of members, and motivation [1].

Of the many approaches, motivation is one of the most difficult to study because it involves the state of an individual's mind. Therefore, we have been investigating a domain model that provides a foundation for motivation-based human resource

[1] 680-4 Kawazu, Iizuka, 820-8502 Japan; E-mail: bashikurume@yahoo.co.jp

management. Typical existing studies on motivation are the content theory by Herzberg [2] and Lawler's process theory [3]. The latter is more suitable for application to our study because it deals with the process of establishing a management method. The model proposed in this theory is known as Lawler's motivation model.

In this paper, we propose a domain model that provides a foundation for motivation-based human resource management. The domain model is composed of three parts. First, an O-I (Organization-Individual integration) model, combining Lawler's motivation model and the IDEF0 [4] process model, is defined as a static model. Then, as a dynamic model on the O-I model, a state transition model is defined to determine the events of management operations and the states of individual members and management. Finally, scenarios for establishing a management method are defined as state transition trails on the state transition model. By applying these models, we analyze an actual example of successfully establishing the CCPM (Critical Chain Project Management) method in a company, and at the same time obtain further scenarios. Thereafter, we discuss primarily the stability of states, motivation, understanding of individuals' roles and their relationship.

Although it has been mentioned that individual motivation should be studied from the organization's viewpoint [5], a detailed study such as ours has, as yet, not been published. Moreover, a precise model for integrating individual motivation and the organization has not been published either. In this paper, Section 2 presents the requirements for domain modeling, while Section 3 describes the proposed domain model. Section 4 analyzes an example of successfully establishing CCPM. A discussion of the domain model and the analysis thereof is given in Section 5.

2. Requirements for Domain Modeling

This section gives an overview of CCPM and describes the requirements for domain modeling.

2.1. Overview of CCPM

The CCPM method [6][7][8][9] is generally well known as a technique for reducing the lead time of a project. The method depicts the dependence between project tasks by means of a network diagram. In traditional project scheduling, each task's duration is estimated with an 80 % HP (Highly Possible) probability of realization. However, in the CCPM method, each task's duration is estimated with a 50% ABP (Aggressive But Possible) probability of realization. To compensate for the ABP probability, a project buffer is placed in the tail of the project to absorb any delay. Therefore, each task does not have a buffer built into its duration as in traditional project management.

In CCPM, a task begins immediately after its preceding task finishes. This contributes to reducing the lead time of the project. Even if some of the project tasks

are delayed, the delay is absorbed by the project buffer and the delivery date is unaffected. Work performance is not measured for each task or individual, but for the whole project or organization. Therefore, there is no concept of cost-performance for each task. CCPM is therefore quite different from traditional project management such as earned value management.

2.2. Requirements for Domain Modeling

Andrews et al. [10] specified three-layers and nine dimensions of business reengineering: process, technology and organization structures in the physical technical layer; reward structure, measurement systems and management methods in the infrastructure layer; and organizational culture, political power and individual belief systems in the value layer. This paper's main themes, namely individual motivation, management methods and project tasks, belong to the value, infrastructure and physical technical layers, respectively. Therefore, our domain model is able to cover all three layers.

Despite being a state of mind, the motivation of individual members is influenced by the company organization through their work. On the contrary, individual performance affects the throughput of the company. Therefore, the mutual dependency of individuals and the company needs to be modeled. One of the main objectives in this paper is to disclose the management operations for establishing a management method in a company. Thus, management operations also need to be modeled. Moreover, the scenarios comprising the operations need to be obtained from the domain model.

3. Domain Modeling

This section proposes a domain model for motivation-based human resource management to establish a management method, namely CCPM. The model is composed of an O-I model, state transition model and scenario generation method.

3.1. O-I model

To determine the mutual dependency of the organization and individuals as mentioned in the previous section, we propose an O-I model that combines the organizational process and individual model. The organization and individual areas determine the organizational process and individual's model, respectively, as shown by the specification of the model in Figure 1. The model is described below with reference to Figure 1.

Lawler's motivation model is applied to *the individual area* as follows. A characteristic of Lawler's motivation model [3] is that the motivation given by expectation is defined as,

$$\Sigma[(E \rightarrow P) \times \Sigma[(P \rightarrow O)(V)] \qquad (1)$$

where effort, performance, outcome and valence are denoted by *E, P, O* and *V,* respectively. *(E→P)* and *(P→O)* denote the expectancy held by the individual members of an organization.

The terms of Equation (1) are defined as follows. *(E→P)* is defined as an individual's belief concerning the probability that if he/she puts effort into performing at a particular level, he/she will be able to perform at that level. *(P→O)* is defined as the individual's subjective probability that performance will lead to an outcome. *(V)* is the valence of the outcome. $\Sigma[(P \rightarrow O)(V)]$ defines the sum of all $(P \rightarrow O)(V)$; each of which is determined for every kind of outcome gained by the performance at a specific level. In other words, this determines an expectation value of the valence for all outcomes of performance at a specific level. $\Sigma[(E \rightarrow P) \times \Sigma[(P \rightarrow O)(V)]]$ means the sum of all $(E \rightarrow P) \times \Sigma[(P \rightarrow O)(V)]$ each of which is determined for every level of performance. This decides the strength of motivation given by expectation.

The strength of motivation defined by Equation (1) is most directly reflected in an individual's *effort,* that is, in how hard the individual works. This effort expenditure may or may not result in good *work performance* since at least two factors determine whether effort is converted into work performance. First, the individual's ability combines multiplicatively with effort to determine work performance. Ability means the comparatively steady and long-range characteristics of individuals such as the nature of their personality, intelligence and skill; thus indicating the individual's power for the work. The second factor, the individual's perception of how his/her effort can best be converted into work performance, that is, role perception in a *problem-solving approach,* also combines multiplicatively with effort to determine work performance. The perception is learned through *observed and personal experience in the situation with stimulation* and *project performance* experience. An outcome is given to project performance. However, Sakashita [5] noted that Lawler's motivation model is itself a general model that supposes various kinds of outcomes. Therefore, we apply the model to various kinds of outcomes. As described in Subsection 2.1, the project performance is not measured for individuals but for the organization as a whole in CCPM. Therefore, the individuals' roles must be wholly managed by the organization.

The organization area is composed of *the company, capacity manager, and client and project layers*. We represent it by using an IDEF0 [4]. In the company layer, *the company management process* constrains *the project processes* using CCPM. In *the capacity manager layer*, the capacity manager *controlling process* assigns *drum resources* to each project process. This is indispensable to the project process. When the project process lacks resources, the company management process allocates additional *non-drum resources* to the project process. In the *client and project layer*, *a client process* passes *specifications* to the *planning process* in the project process. The *planning process* makes a project p*lan* to constrain the *executing process* and project manager *controlling process*. The *planning process* also passes *specifications* to the

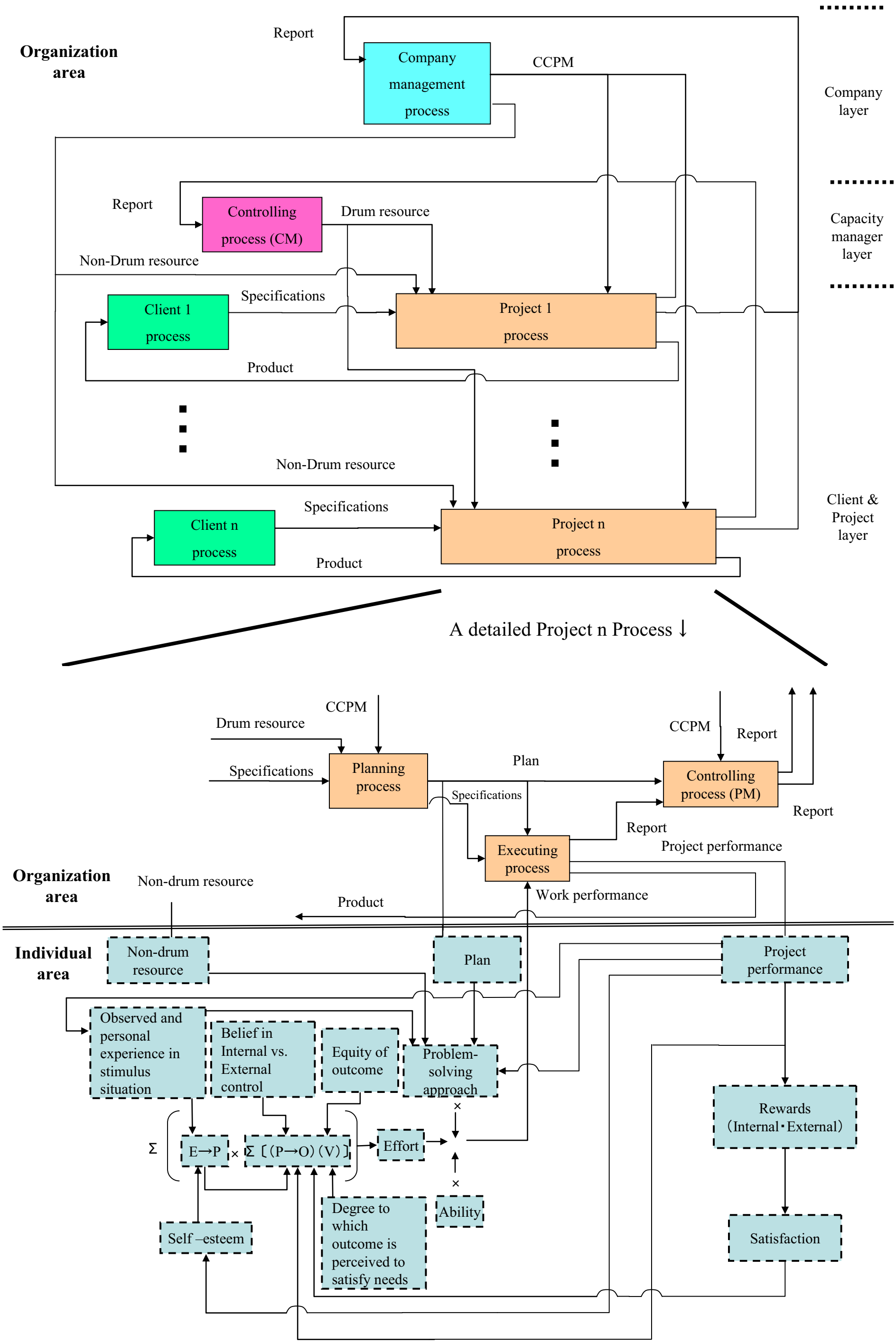


Figure 1. O-I model

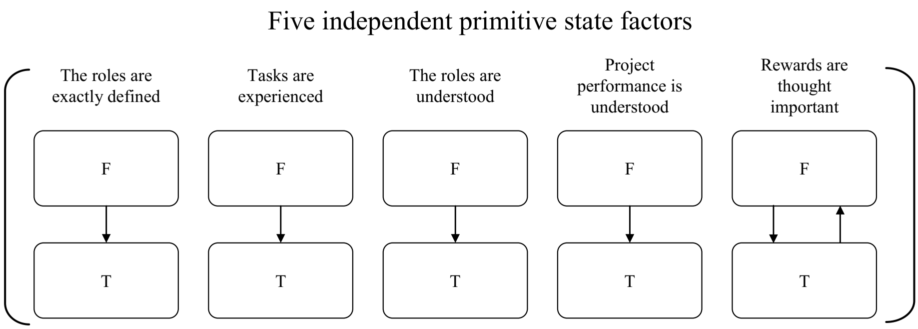
executing process. After completing the project, the *executing process* passes *product* to the *client process*. The *executing process* also submits a *report* to the *controlling process*. This *report* is in turn passed on to the *company management process* and capacity manager *controlling process*. Performance, including experience, output from the executing process, is input into the individual area. The problem-solving approach in the individual area receives the plan from the planning process, and a non-drum resource from the company management process.

3.2. Dynamic Model

The organization undertakes various operations in the process of introducing and establishing a management method. Then, the organization chooses an operation to be undertaken the following time depending on the current state of the organization. In order to determine each set of states or operations, we define a dynamic model, that is, a state transition model.

States of the organization in the O-I model described above, can be represented by the Cartesian product of the states of five independent primitive state factors. The five primitive factors are specified in Figure 2. For example, the leftmost factor in the figure means that the roles of all members in the organization are exactly defined. The factor takes one of two states: true or false denoting, respectively, that the roles are exactly defined or not. The other four primitive factors are defined in a similar way.

Each primitive factor moves from one state to another by following an arrow specified in the figure. The state transition occurs as a result of an event. Here, each primitive factor has its own event for state transition. One or some of the events of the primitive factors can occur simultaneously. Therefore, the state transitions determined by the model are non-deterministic.



States of an organization are represented by the Cartesian product of the states of five independent primitive state factors. T: True F: False

Figure 2. State transition diagram

By the way, an event is not specifically defined in our model. For example, the middle factor in the figure has an event to move from the state that the roles are not understood to the state that the roles are understood. This state transition can be initiated by one of two operations: meeting or training. The specific implementation of the operation for the event is left to the organization.

3.3. Scenario Generation

A series of consecutive state transitions represents a scenario [11], from introducing to establishing a management method. Of course, specific operations are assigned to events in the scenario. Figure 3 shows an example of a scenario, obtained from the state transition model specified in the figure.

As described in the previous subsection, our state transition model is non-deterministic. Moreover, the specific implementation of the operation of the event is left to the organization. These two features are useful for generating various scenarios.

Scenario / State factors	(1) → Before kickoff meeting	(2) → Kickoff meeting	(3) → Experience of work (4) 1st meeting	(5) → 2nd meeting	(6) → 3rd meeting (7) 4th meeting	(8) Established CCPM
The roles are exactly defined	**T**	**T**	**T**	**T**	**T**	**T**
Tasks are experienced	**F**	**F**	**T**	**T**	**T**	**T**
The roles are understood	**F**	**F**	**F**	**T**	**T**	**T**
Project performance is understood	**F**	**F**	**F**	**F**	**T**	**T**
Rewards are thought important	**F**	**T**	**T**	**T**	**T**	**F**

Figure 3. A generated scenario

4. Applying the Model

A bridge design company successfully established CCPM. The company carried out an anonymous survey via a questionnaire asking its members “what do you think of the newly introduced CCPM method?”. The company held 4 meetings from the time the company introduced CCPM until it was established. The meetings were held in September 2004, October 2004, November 2004, and finally in April 2005. No survey was conducted at the third meeting; instead a review was carried out. Moreover, a year and 4 months later, a 5th questionnaire was introduced.

Answers to the first questionnaire showed much dissatisfaction with the project. Typical answers were “We don’t know what we should do next.” and “Because the critical chain has not been shown to us, we don’t understand the task or why we helped.” These answers imply failure in the perception of an individual’s role in the problem-solving approach of the individual area in Figure 1. However, the individuals are obliged to continue working at the performance level required to carry out their projects. The level is determined by the multiplication of the problem-solving approach, effort and ability as specified in the individual area. Here, individual ability is assumed to be fixed for each individual in the short term. Therefore, an individual must apply additional effort to compensate for the failure in role perception in the problem-solving approach. Moreover, because they were not informed of the outcome of the work performance at the introduction of the CCPM method, they received no internal reward. After analyzing this questionnaire, the company clarified the roles of the project managers and members, and explained these to the individuals.

The answers to the second questionnaire also showed much dissatisfaction. Answers given include “Who communicates with a customer about each order?” and “Do project managers know the status of all orders?” However, the dissatisfaction was not related to the perception of failure in their own roles, but in the roles of others in the organization. In other words, the perception of their own roles was strengthened by the first meeting. However, the perception of the roles of others was not yet adequate in that they had not yet completely perceived the problem-solving approach. This required additional individual effort, and resulted in dissatisfaction. After this questionnaire, the company explained the individual roles again to strengthen the perception of the problem-solving approach.

At the third meeting, the company announced an initial outcome of CCPM. Specifically, a project had been completed with about a 1/3 – 1/2 reduction in the planned lead time. Thus, the individuals perceived the outcome of the project performance in their individual area and were satisfied.

The answers to the fourth questionnaire showed increased satisfaction; for example, “The work has become more efficient because overtime has decreased compared to last year.” and “We don’t have to do two or more tasks at the same time, because we make a plan before working.” Moreover, they demanded a better way of working and

education; for example, "Overtime has decreased, but some people are still working until midnight as before." and "We need to be informed of resource assignments with respect to all projects." In other words, they were satisfied because their perception of the problem-solving approach increased in relation to their work experience. They also received the internal reward related to the success of introducing CCPM and the external reward of decreasing their overtime. On the other hand, the increased satisfaction decreased their awareness of the importance of the reward. Consequently, their motivation and effort decreased. However, the work performance level required to carry out their projects could be preserved with ordinary effort, because the problem-solving approach had already been strengthened. In other words, CCPM had been established.

The answers to the fifth questionnaire contained no comments on dissatisfaction, only satisfaction; for example "Projects turn out as planned and design errors have decreased." Of course, there were some comments on how to improve the treatment of special works. Thus, CCPM was already established.

5. Discussion

This section discusses the usefulness and features of the domain model proposed in this paper and describes future research. As described in the previous section, the actual specific operations undertaken by the company correspond to the state transitions that are defined in the domain model. The significance of the operational events, such as holding meetings, has been approved by the corresponding transitions of important states, such as individuals being aware of their roles and project performance. Therefore, the domain model would be useful in analyzing actual examples of establishing CCPM.

The state transition model has two features useful for generating various scenarios. One is that a state transition is non-deterministic. That is, a state can move to a different state as a result of different operational events. The different operational events generate different scenarios. The other feature is that the specific implementation of event as operation is left for organization. Therefore, the same transition can happen as a result of different specific operational events. The different specific operational events also generate different scenarios.

The state of an organization, including individual motivation, must be stable for a certain period to implement a scenario in the organization. One of the most serious obstacles to the stability of states is the excessive individual effort required by the organization. Investigating the stability of states is an on-going study.

In the future, we intend studying evaluation criteria for choosing a scenario suited to the current state of the organization. To achieve this, we need to learn the content theory of motivation and integrate it into our domain model. As many companies are currently engaged in introducing new management methods, we should tailor our

domain model to the methods and apply them. Moreover, the usefulness of the domain model needs to be evaluated.

6. Conclusion

We have proposed a domain model for establishing CCPM and have discussed the application thereof in the analysis of an actual example. Through this study, we have understood that the “problem-solving approach” and “experience” are very important in establishing a management method. Moreover, we need to continue investigating the stability of states.

In the future, we aim to investigate evaluation criteria for choosing a scenario, integrate the content theory of motivation into our domain model, tailor our domain model to new management methods and apply these.

Acknowledgement

The work described in this paper was partially supported by Grants-in-Aide for Scientific Research (No. 18500025) of the Japan Society for the Promotion of Science.

References

[1] Konosu, T., “Special issue on Human Factors management”, Journal of the Society of PROJECT Management, Vol.6, No.2, 2004 (in Japanese).

[2] Herzberg, F., “Work and the Nature of Man”, World Publishing, 1966.

[3] Lawler E.E., “Pay and Organizational Effectiveness A Psychological View”, New York, McGraw-Hill, 1971.

[4] Bernus, P., Mertins, K., Schmidt, G., “Handbook on architectures of information systems”, Springer, 1998.

[5] Sakashita, S., “Organization action research”, Hakutoushobou, 1985(in Japanese).

[6] Goldratt, E. M., "Critical Chain - US", North River Pr, 1997.

[7] Goldratt, E. M.,"It's Not Luck - US", North River Pr, 1994.

[8] Nakajima, H., Tsumagari, K., “Project management Critical Chain”, Nihonnouritsukyoukai management center, 2003 (in Japanese).

[9] Tsumagari, K., “I can do that, Critical Chain”, Diamond, 2004.

[10] Andrews, D.C., Stalick, S.K., ”Business Re engineering -- the survival guide”, Youdon Press, 1994.

[11] Ohnishi, A., “A Generation Method of Exceptional Scenarios form Normal Scenario” IEICE TRANSACTIONS on Information and Systems Vol.E91-D No.4, 2008.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-42

Dependency of intentions on i*

Shuichiro Yamamoto
Research Institute for System Science, Research and Development Head Quarters, NTTDATA, Tokyo, JAPAN

Abstract. The Strategic Dependency Model (SD model) of i* framework is used to model intentions among actors. However, it is not clear how to validate the equivalence of SD models. In this paper, the dependency matrix of intentions is proposed to formalize equivalence of SD models. We also provide a method to develop SD models from dependency matrices. The proposed method can be used to reduce redundant actors using dependency of intentions.

1. Introduction

The i* framework is a popular goal oriented requirements analysis method and provides the SD model to analyze intentions among actors [1] [2]. However, it does not provide the method to analyze the equivalence of SD models and reduction of the complexity of relationships among actors. If we try to decide the goodness of the SD models, it is necessary to define the equivalence and comparison method for SD models.

This paper is organized as follows. First, the goal graph structure of SD model is introduced. Then the basic concept of Intention Dependency matrix (ID matrix) is explained. Next, a method to develop SD model is shown based on the ID matrix. At last the Intention Dependency algebra (ID algebra) is also proposed. The relationship between ID matrix and ID algebra is shown. Issues of the proposed method are mentioned for conclusion.

2. SD model of i* framework

The SD model S is defined as <Actors, Intentions, Dependency>. The actor dependency is defined as the triple tuple (depender, dependum, dependee) where depender and dependee are actors. The dependum is the intention that depender expects it to dependee.

Figure 1 shows an example of SD model S1= < {A,B,C,D}, {P,Q,R}, {(A,R,B), (B,Q,C), (C,P,B), (C,R,B),(C,Q,D), (D,P,A) }>

3. Intention Dependency Matrix for i*

[Definition 1] Intention Dependency matrix

Intention Dependency matrix M for SD model S=<A, I, D> is defined as follows.
M (n, m) = 1 if D∍(a, n, b), (b, m, c), A∍a,b,c, I∍n,m, n≠m
M (n, n) = 0 for I∍n

Figure 2 shows an example of ID matrix M for the SD model of Figure 1.

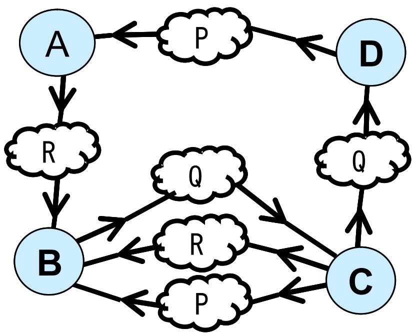


Figure 1 Example of SD model

	P	Q	R
P	0	1	1
Q	1	0	1
R	0	1	0

M_1

Figure 2 Example of ID matrix

We can now define the equivalence of SD models using ID matrix.
[Definition 2]
SD model S and S' are equivalent if δ (S) = δ (S'). Where δ (S) and δ (S') are the ID matrices for S and S' respectively.

4. SD model development method based on ID matrix

An SD model S is developed from ID matrix M using the following steps.
[Method] SD model development
(STEP1) Actors of SD model M are prepared for each rows of M.
(STEP2) For each actor X_R, the corresponded intention is assumed as R. Then (X_P, R, X_R) is added as the element of Dependency set of S, where M (P, R) =1 and P is the corresponded intention of X_P.

Figure 3 shows the generated SD model from ID matrix M_1 of figure 2. The SD model includes Dependency set {(X, P,Y),(X,R,Y),(Y,Q,X),(Y,R,Z),(Z,Q,Y)}.

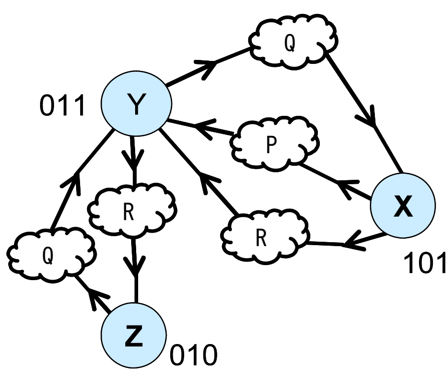


Figure 3 SD model for ID matrix of Fig.2

In the Figure 3, X, Y and Z are corresponded to rows (101), (011) and (010) of ID matrix of figure 2 respectively. The SD model shown in Figure 3 reduces the number of actors of the SD model in Figure 1.

5. Intention dependency algebra of SD model

The ID matrix defined in the above section implicates the algebraic expression of intention dependencies.

[Definition 3] Intention dependency term

(1) The intention P is an intention term.

(2) $P \rightarrow Q$ is an intention dependency term when P and Q are intentions.

[Definition 4] Intention dependency list

(1) The empty is the special case of intention dependency list.

(2) The intention dependency term P is an intention dependency list.

(3) P; Q is an intention dependency list when P and Q are intention dependency lists.

[Definition 5] Intention dependency of SD model

For the SD model S=<A, I, D>, an intention dependency list L is constructed as follows.

(1)Let L be an empty list.

(2) For each pair of intentions n and m, if $D \ni (a, n, b), (b, m, c)$, $A \ni a,b,c$, $I \ni n,m$, $n \neq m$ then $n \rightarrow m$ is appended to L.

For example, the following ID lists L_1 and L_2 are constructed from figure 1 and 3 respectively.

$$L_1 = P\rightarrow R;R\rightarrow Q;P\rightarrow Q;Q\rightarrow P;Q\rightarrow R;Q\rightarrow P;$$

$$L_2 = Q\rightarrow R;P\rightarrow Q;P\rightarrow R;R\rightarrow Q;Q\rightarrow R;Q\rightarrow P;$$

[Definition 6] Transformation rule

(TR1) $V;W \Rightarrow W;V$

(TR2) $V;V \Rightarrow V$

For example, we can reduce L_1 using the transformation rule as follows.

$L_1 \Rightarrow P\rightarrow R;R\rightarrow Q;P\rightarrow Q;Q\rightarrow R;Q\rightarrow P;Q\rightarrow P;$ (by TR1)

$\Rightarrow P\rightarrow R;R\rightarrow Q;P\rightarrow Q;Q\rightarrow R;Q\rightarrow P;$ (by TR2)

$\Rightarrow P\rightarrow R;P\rightarrow Q;R\rightarrow Q;Q\rightarrow R;Q\rightarrow P;$ (by TR1)

In the same way, we can reduce L_2 as follows.

$L_2 \Rightarrow P\rightarrow Q; Q\rightarrow R;P\rightarrow R;R\rightarrow Q;Q\rightarrow R;Q\rightarrow P;$ (by TR1)

$\Rightarrow P\rightarrow Q; P\rightarrow R; Q\rightarrow R;R\rightarrow Q;Q\rightarrow R;Q\rightarrow P;$ (by TR1)

$\Rightarrow P\rightarrow Q; P\rightarrow R; R\rightarrow Q; Q\rightarrow R;Q\rightarrow R;Q\rightarrow P;$ (by TR1)

$\Rightarrow P\rightarrow Q; P\rightarrow R; R\rightarrow Q; Q\rightarrow R;Q\rightarrow P;$ (by TR2)

$\Rightarrow P\rightarrow R; P\rightarrow Q; R\rightarrow Q; Q\rightarrow R;Q\rightarrow P;$ (by TR1)

[Definition 7] Algebraic interpretation of ID matrix

The algebraic interpretation for the ID matrix M is the ID list constructed from M. We can assume each row of M is corresponded to an intention of SD model S=<A, I, D>.

If M (n, m) = 1 then $n \rightarrow m$ is extracted as an element of algebraic interpretation where $I \ni n, m$, $n \neq m$.

The algebraic interpretation of M is finally defined as the connection of every extracted element using the symbol ";".

For example, the algebraic interpretation $\pi(M_1)$ of ID matrix of figure 2 is as follows.

$$\pi(M_1) = P\rightarrow R; Q\rightarrow P; Q\rightarrow R;R\rightarrow Q; P\rightarrow Q;$$

We now can show the following properties on the intention dependency

representations define above.

[Property 1] Algebraic equivalence

For SD model S and S', S is equivalent to S' if and only if π (M) = π (M'). Where M, M' are their ID matrices respectively.

(Proof)

First, if S is equivalent to S' then their ID matrices are also equivalent. Then δ (S) = δ (S'). Therefore the algebraic interpretations are also equivalent. π (δ (S)) = π (δ (S'))

Next, we assume π (M) = π (M') conversely. If δ (S) $\neq$ δ (S'), there is an element M(P,Q) $\neq$ M'(P,Q). This contradicts to the assumption π (M) = π (M').

Therefore δ (S) = δ (S').

[Property 2] Algebraic transformation

For SD model S, S' and their ID lists θ (S), θ (S') ,

M is equivalent to M' if and only if there is a transformation from θ (S) to θ (S').

(Proof)

First, if M is equivalent to M' then their ID matrices are also equivalent. Thus δ (S) = δ (S') holds. If there is no transformation from θ (S) to θ (S'), there is an element P$\rightarrow$Q in θ (S) and P$\rightarrow$Q is not in θ (S'). If P$\rightarrow$Q is in θ (S) then (a, P, b) and (b, Q, c) are in D. And if P$\rightarrow$Q is not in θ (S') then (a, P, b) and (b, Q, c) are not in D'. This contradicts δ (S) = δ (S'). Therefore θ (S) and θ (S') have the same set of elements. This means there is a transformation from θ (S) to θ (S').

Next, we assume there is a transformation from θ (S) to θ (S'). It is clear that S and S' have the same ID matrix. Otherwise, δ (S) and δ (S') have an element M(P,Q) $\neq$ M'(P,Q). This also contradicts that θ (S) can be transformed to θ (S').

6. Conclusion

In this paper, an actor reduction method for i* framework based on the intention dependency is proposed. It also proposed the ID matrix and ID algebra using the intention dependency. Moreover, the equivalence of SD models is clarified and used to show the relationship between ID matrix and ID algebra. Although the goal reduction method for the goal graphs of NFR framework was proposed in [3], actor reduction methods have not been proposed yet. The proposed method will contribute to automate goal oriented requirements engineering process.

It is also necessary to examine the effectiveness of the method for the practical information system development.

References

[1] Lamsweerde, A., Goal-Oriented Requirements Engineering: A Guided Tour, RE'01, pp.249-263, 2001.

[2] Shuichiro YAMAMOTO and Haruhiko KAIYA, Karl Cox, and Steven Bleistein, Goal Oriented Requirements Engineering -- trends and issues -, No.11, pp.2701-2711, IEICE 2006.

[3] Shuichiro Yamamoto, Goal Algebra for IT Alignment, proc. of JCKBSE 2006, pp.147-150.

This page intentionally left blank

Development Processes for
Knowledge-Based Applications

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-49

Stratified Composition of Web Services

Riina MAIGRE [a], Pavel GRIGORENKO [a], Peep KÜNGAS [b] and Enn TYUGU [a]
[a] *Institute of Cybernetics, Tallinn University of Technology, Estonia*
[b] *Royal Institute of Technology (KTH),*
School of Information and Communication Technology, Sweden

Abstract. We propose an automatic service composition methodology where, three levels of composition knowledge are distinguished: user level, logical level and implementation level knowledge. We use a knowledge-based software development environment CoCoViLa that enables composition throughout these three levels. A motivation for this approach is a need to overcome the complexity of service composition on very large sets of atomic services we are dealing with in our application domain. The domain concerns federated governmental information systems.

Keywords. automatic service composition, knowledge-based software development, federated information systems.

Introduction

In this article we propose an automatic service composition methodology and describe a tool supporting it. The methodology distinguishes clearly three knowledge levels: user level, logical level and implementation level. The novelty of our work is stratification of knowledge usage into layers. This enables us to develop a tool that hides most of the complexities of service composition from its users, allowing a user to communicate in a visual or textual mode using only the domain ontology. Also internal usage of knowledge is stratified. This separates logic from implementation and facilitates both synthesis of an algorithm of a service and generation of the final description of the service. We use a knowledge-based software development environment CoCoViLa [1] that supports these levels. A motivation for this approach is a need to overcome the complexity of service composition on very large sets of atomic services that we are dealing with. More specifically, we are considering composition of services for federated governmental information systems which are providing public services in Estonia. The proposed methodology and the accompanying tool for automation of service composition can be used by software developers to reduce their time and other resources for maintenance and extension of an existing governmental information system.

We have surveyed methods applied in automated composition of services. Theoretically any domain-independent AI planner can be applied for Web service composition. In [2] SHOP2 planner is applied for automatic composition of DAML-S services. Other planners, which have been applied for automated Web service composition, include [3,4,5,6].

However, we need a composition method that will work efficiently for a very large set of atomic services, and the AI planners found, do not satisfy this requirement.

Namely, traditional AI planners are optimised for problems with deep search tree and low branching factor versus shallow but broad search space, which is the case with Web service composition.

Some other methods of automated composition of services are inspired by the works of software synthesis. Waldinger [7] proposed initial ideas for a deductive approach for Web services composition. The approach is based on automated deduction and program synthesis and has its roots in the work presented in [8]. Initially available services and user requirements are described with a first-order language, related to classical logic, and then constructive proofs are generated with Snark [9] theorem prover. From these proofs workflows can be extracted.

In [10] a modification of Golog [11] programming language is used for automatic construction of Web services. Golog is built on top of situation calculus and has been enriched with some extra-logical constructions like *if*, *while*, etc. Golog also provides an efficient way to handle equivalence relations. Therefore, it is argued that Golog provides a natural formalism for automatically composing services on the Semantic Web.

McDermott [12] tackles the closed world assumption in AI planning while composing Web services. He introduces a new type of knowledge, called value of an action, which allows modelling resources or newly acquired information – entities, which until this solution were modelled extra-logically. Anyway, while using resource-conscious logics, like linear logic, applied by Rao et al [13], or transition logic, this problem is treated implicitly and there is no need to distinguish informative and truth values. Since linear logic is not based on truth values, we can view generated literals as references to informative objects.

Hull and Su [14] present a short overview of tools and models for Web service composition. The models include OWL-S, the Roman model [15] and the Mealy machine [16]. While OWL-S includes a rich model of atomic services and how they interact with an abstraction of the "real world", the Roman model and the Mealy machine use a finite state automata framework for representing workflows.

Hashemian and Mavaddat [17] combine breadth-first graph search and interface automata [18] for automating Web service composition. While graph search is used for finding a path with minimum length from identified input nodes to identified output nodes, interface automata is applied for composing paths into a composite Web services. Graph search operates over a directed graph, where edges represent available Web services and nodes represent inputs/outputs of particular Web services.

In the surveyed works we did not find a method scalable for our case where there are more than a thousand atomic services that are all candidates for inclusion in the composed service and there is insufficient semantic information for filtering out a smaller set of candidate services. We have decided to use a software development environment CoCoViLa [1] and to develop on its basis a package for service composition on large sets of atomic services. CoCoViLa is a programming environment that supports automatic (deductive) synthesis of programs and generates Java code from visual and textual specifications. Its advantages include extensibility and its ability to synthesize programs from specifications containing descriptions of thousands of atomic computations (given by means of equations or methods of predefined classes) that are candidates for inclusion in the composed program.

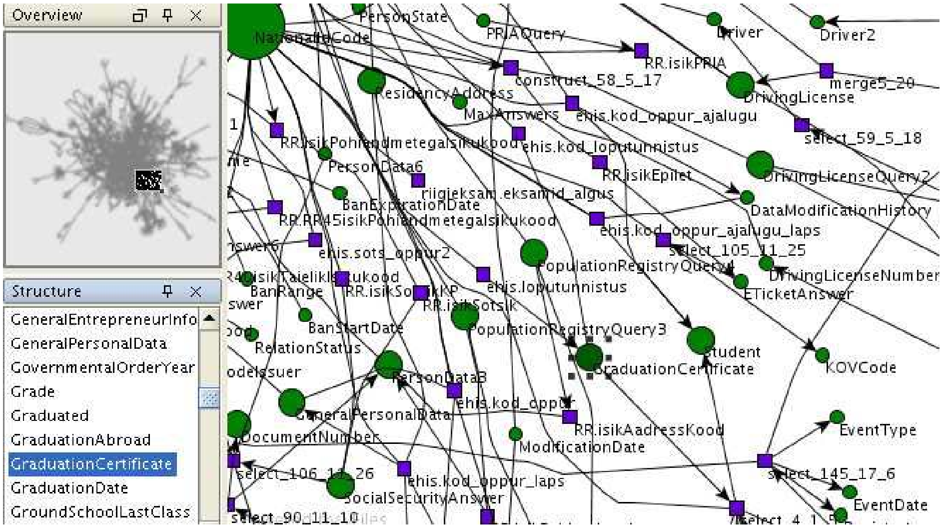


Figure 1. Atomic services of the e-government information system.

1. Example application

Experiments referred in this paper have been performed on a part of the federated e-government information system [19], complying with the service-oriented architecture (SOA), that has been developed in Estonia during the recent years. Development of the federated information system has been a complicated task that has required cooperation between government agencies that already provide services for citizens.

Analysis of the system has resulted in a unified service model that contains about three hundred atomic services, including a number of rather primitive data transformers needed for interoperability of databases. Totally more than a thousand atomic services are available, which can be included to the service model and composed into complex services. However, this large system is continuously changing, and thus maintenance of its model requires also continuous efforts.

A complexity of the set of services involved in the e-government information system depends not only on the number of atomic services, but also on the number of relations between them. To give an idea of this complexity, we present a part of these services (about 300 out of totally more than a thousand services) as a graph in the Java graph editor yEd window in Fig. 1. The figure shows a graph where square and round nodes represent atomic services and resources respectively. Size of a circle of a resource shows its relative importance (connectivity to services). A small window on the left side shows the general view of the graph. Another window on the left side includes a scrollable list of all resources. A small part of the model is enlarged in the right window. A resource with the largest value in the current model is *NationalIdCode* that is not surprising, because it is used in the majority of services.

The central part of the Estonian e-government information system is its infrastructure, called X-road, guaranteeing secure access to national databases over the Internet [19]. X-road enables secure access to nearly all Estonian national databases whose services are available through domain-specific portals to a variety of user groups (citizens,

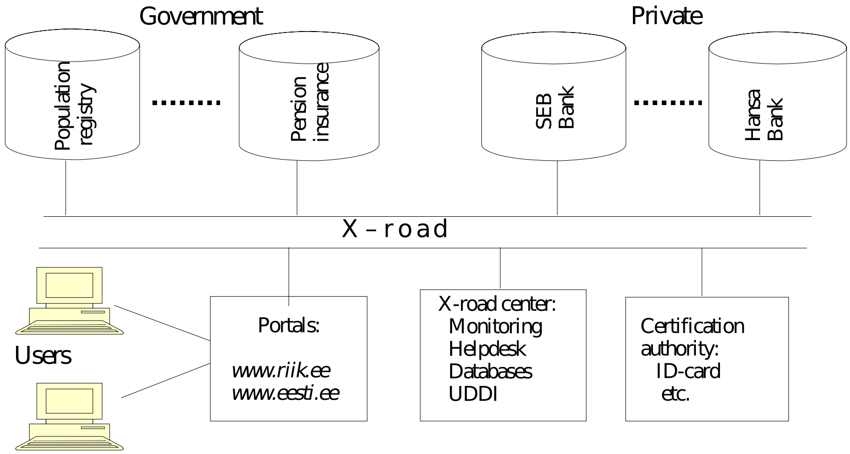


Figure 2. X-road connects public and private information service providers.

officials, entrepreneurs). All Estonian residents having the national ID card can access these services through X-road. The number of requests per month exceeds currently 3 million and is still growing. For the sake of brevity we are going to call the whole information system from now on as X-road.

Fig. 2 shows the structure of the national information system based on X-road. As demonstrated in the figure, X-road connects, besides government databases, also some private databases. These are, for instance, main banks and some privately owned infrastructure enterprises. Some services are provided by the X-road infrastructure itself – that includes PKI infrastructure, registries, helpdesk etc.

Users connect to the system through a variety of portals. They use predefined services implemented by software developers. The set of available services is expanding continuously. Until now, a user service is composed of atomic services of a single public authority or enterprise, e.g. immigration office, bank etc. However, X-road allows one to unite services of different authorities and enterprises into a compound service, and our goal is to develop a tool for helping a software developer to do this.

To demonstrate the composition of a service, let us consider the following example, where an official has to find out person's contact address. Different databases of X-road registries have their own copy of person's address, which may not be the same in all registries. Therefore, to find all possible addresses of a particular person, an official has to query all databases requesting the address as an output. Currently these queries have to be executed separately on different databases. A complex service would allow inserting an Estonian ID code or another identifier as an input to get different contact addresses as an output.

To create this complex service the developer has, first, to find relevant data items of the service model. In our example these items are *NationalIdCode* as an input and *AddressString*, *EstonianAddressString*, *OwnerAddressString*, *ResidencyAddress*, *ResponsibleUserAddress* as outputs. Marking these variables as inputs and outputs in the specification, the developer gives a goal for the synthesizer. The following steps will be performed fully automatically and, if it is possible to find a sequence of atomic services that produce required outputs from the inputs, then the composite service is constructed and its description in BPEL will be generated.

2. Knowledge architecture of the stratified service composition tool

In this section we describe knowledge architecture of our service composition tool, ignoring its implementation details. To describe the knowledge architecture we are going to use the definition of a *knowledge system* (KS) and the notations introduced in [20]. A knowledge system is a component – a module of knowledge architecture of a knowledge-based system. It includes a knowledge language, a knowledge handling mechanism, e.g. an inference engine, and a method for associating meanings to knowledge objects represented in the knowledge language. We assume that there is always a set of meanings of a knowledge system and a mapping from knowledge objects into the set of meanings. We are going to depict explicitly the set of knowledge objects *S* and the set of meanings *M* of a knowledge system in visual representations of knowledge architecture as shown in Fig. 3.

Figure 3. Notation of a knowledge system

A knowledge system can be formalized as an interpreted free deductive system [21]. However, in our case we are not using this formalization. Knowledge systems can be composed into larger knowledge architectures by connecting them hierarchically, semantically, operationally etc. Fig. 4 shows notations for hierarchical and semantic connections of knowledge systems. Two knowledge systems K_1 and K_2 with sets of notations S_1, S_2 and sets of meanings M_1, M_2 respectively are *connected hierarchically*, if there is a relation *R* between the sets M_1 and S_2, i.e. meanings of the knowledge system K_1 tell something about the knowledge objects of the knowledge system K_2. K_1 is called upper and K_2 is called lower system. If the relation *R* is one-to-one mapping between some subsets of M_1 and S_2, then we say that the knowledge systems are *strongly hierarchically connected.*

Knowledge systems K_1 and K_2 are connected semantically, if they have one and the same set of meanings *M*. This is shown in Fig. 4 (b).

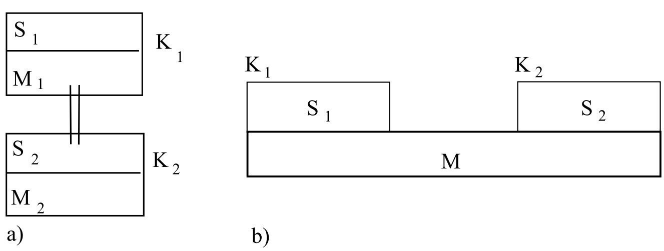


Figure 4. Hierarchical (a) and semantic (b) connection of knowledge systems

Our tool has three knowledge levels:

- *user knowledge level* with visual and textual representation of knowledge;
- *logical level* for formal representation of knowledge and automatic composition of new services;

- *service implementation level* for grounding of services – calling services and performing actual computations.

Fig. 5 shows knowledge architecture of the service composition tool developed as a package in CoCoViLa. On the user knowledge level there are two semantically connected knowledge systems: visual and textual knowledge systems. These systems have one and the same set of meanings – unfolded (detailed) specifications of services. This allows us to apply union operation and to get one user knowledge system. Meanings of knowledge objects are unfolded textual representations of specifications on this level. Their elements are almost in one-to-one correspondence with knowledge objects of the logical level. This makes it easy to introduce the hierarchical connection between the user knowledge and the logical level. Knowledge objects on the logical level are logical formulas representing functionality of atomic services and data structures. This level is hidden from the user. Meanings of logical level are algorithms that are realizations of intuitionistic formulas. These algorithms are mapped into Java programs on the service implementation level. Finally, meanings of the service implementation level are composed service descriptions generated by Java programs. The following sections describe the three knowledge levels in more detail.

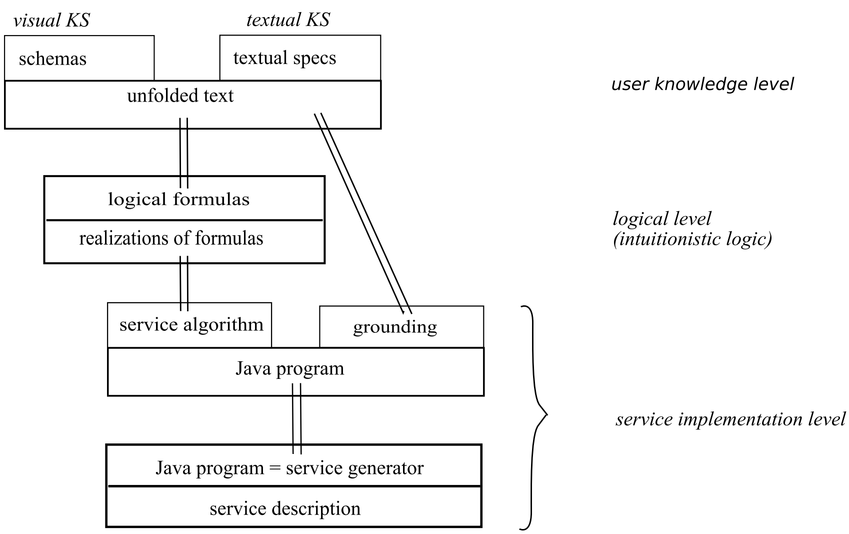


Figure 5. Knowledge architecture of the service composition tool

3. User knowledge-level

The user knowledge level supports visual and textual representation of knowledge about services and syntactic service models, handling of inheritance, equality and structural relations. Knowledge objects are both visual and textual specifications of service composition problems given by users. This level is implemented in CoCoViLa by extending it with suitable metaclasses for a generic service. Inheritance, equality and structural relations are supported entirely by CoCoViLa. One can draw a visual representation of a

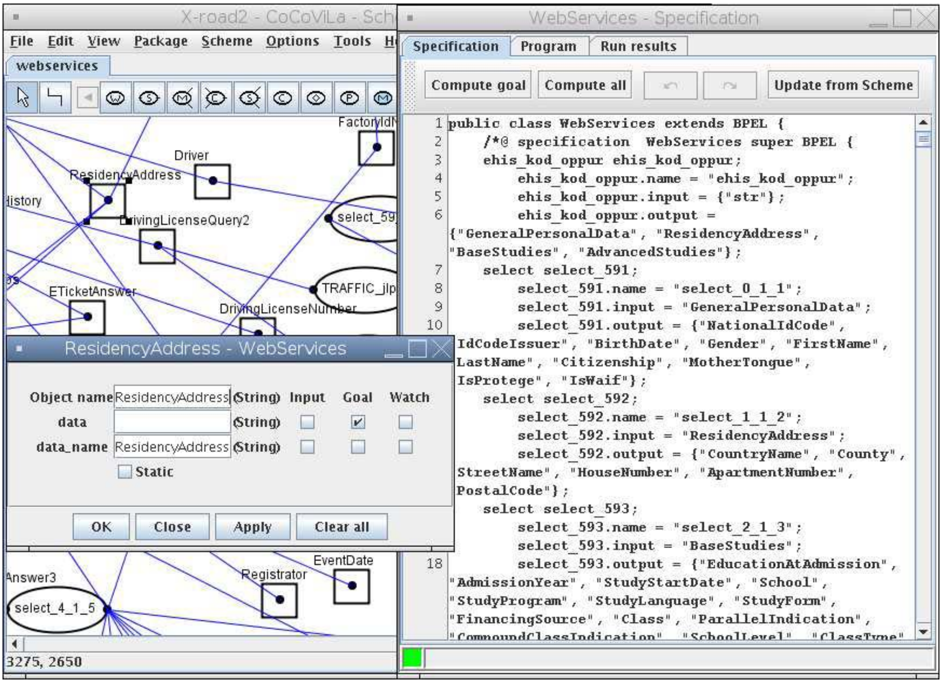


Figure 6. User knowledge level

syntactic model of services, introduce values of attributes of components of the model and specify a goal that must be reached by the expected service. A screenshot in Fig. 6 shows visual and textual representations of knowledge on the user knowledge level. The left window is for visual description of services. Its toolbar shows different visual components for atomic services. The scheme in it corresponds in principle to the graph of atomic services represented in Fig.1. It can be even generated automatically from the latter. A goal of the service to be composed is defined in the pop-up window. The right window shows a text that has the same meaning as the scheme on the left.

4. Logical level

Knowledge system of the logical level can be described very precisely in two different ways. It can be represented as an implicative part of the intuitionistic propositional calculus [22]. A comprehensive description of representations of services on the logical level has been published in [23]. The formulas describing in CoCoViLa inputs and outputs of services become implications on the logical level and are handled by a very efficient theorem prover. The knowledge objects on the logical level are logical formulas – showing the functionality of atomic services. Meanings of logical formulas on this level are algorithms. A new composite service is synthesized by means of a deductive program synthesis method called structural synthesis of programs. (This method is a part of CoCoViLa functionality, see for instance [24].)

In principle the synthesis process is as follows. The input to the process is a set of logical formulas that represent knowledge about the atomic services that can be used as

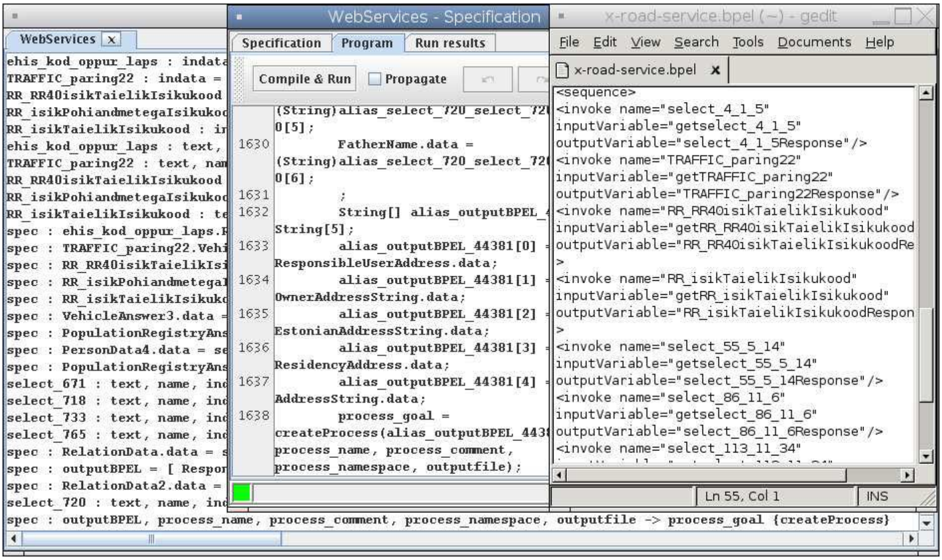


Figure 7. Service implementation level

components of the synthesized service, and also a goal, i.e. a formula that describes the wanted service.

The tool tries to derive logically the goal. If this is possible, then the service can be composed – meaning of the derived goal is the algorithm of the expected service. This algorithm is represented only by its structure. Implementation details are available from specification given on the user knowledge level, see relation between the unfolded text and grounding in Fig. 5.

5. Service implementation level

The main input of the service implementation level is a representation of the structure of a synthesized service received from the logical level. Another input comes from the specification, see Fig. 5. These inputs together have meanings as Java programs. Output of this level is a specification of the service in BPEL. Looking closer at this level, we see that it is a hierarchical connection of two knowledge systems. The higher one takes structure of a service and grounding of atomic services, and produces a meaning that is a Java code with a method call for each atomic service included into the composed service. The lower one takes the Java code as a knowledge object and produces a service description as the respective meaning. Speaking in software terms– first a generator for generating a composed service is synthesized, and thereafter it is run and a required service description is generated.

Fig. 7 is a screenshot with three windows showing knowledge objects on service implementation level. The leftmost window is the algorithm of a service to be composed. This is an input for the service implementation level. The central window is a Java code that corresponds to the algorithm. However, this is not a code of a service, but by compiling and running this code the service description in BPEL is generated that is visible in the window on the right side.

The variable *process_goal* on the right hand side of the implication in the last line of the service algorithm in Fig. 7 shows that the goal of service composition is finally reached by running the method *createProcess* (shown in curly brackets). The inputs of this step of the algorithm are a text in BPEL that has been generated already – (*outputBPEL*), some general information about the service (*process_name*, *process_comment*, *process_namespace*) and a name of the file where the output goes (*outputfile*). This step is also the last method call in the Java program that is visible in the program window in Fig. 7.

6. Concluding remarks

Our aim has been to develop a complex knowledge-based tool with stratified knowledge levels. This paper describes the architecture of the tool without explaining its implementation details. The latter requires the usage of a concept of knowledge system as an architectural component of software. Implementation details can be added to the architectural description as a proof of realizability. The last two figures of the present paper have this role – they explain the operation of the tool in terms of inputs and outputs of architectural components and rely on a real case of synthesis of a service on a very large set of atomic services. In the restricted space of the paper, we are unable to describe the composition algorithm that handles knowledge on the logical level. This description is available in [24].

The main features of the presented architecture are

- user friendly upper level of the hierarchical connection of components consisting of semantic connection of visual and textual knowledge systems for handling domain knowledge;
- precisely defined mapping of meanings of user level to knowledge objects of logical level;
- precisely defined relation between the meanings of logical level and knowledge objects of implementation level.

It is interesting to notice that the latter relation is known in logic as the Curry-Howard correspondence between logical formulas and a representation of computable functions.

Acknowledgements

This work has been partially supported by the Estonian Science Foundation grant No. 6886.

References

[1] P. Grigorenko, A. Saabas, and E. Tyugu. Visual tool for generative programming. *ACM SIGSOFT Software Engineering Notes*, 30(5):249–252, 2005.

[2] D. Wu, B. Parsia, E. Sirin, J. Hendler, and D. Nau. Automating DAML-S Web services composition using SHOP2. In *Proceedings of the 2nd International Semantic Web Conference (ISWC 2003)*, 2003.

[3] M. Pistore, F. Barbon, P. Bertoli, D. Shaparau, and P. Traverso. Planning and monitoring web service composition. In *Proceedings of the 11th International Conference on Artificial Intelligence, Methodologies, Systems, and Applications (AIMSA 2004)*, volume 3192 of Lecture Notes in Computer Science, pages 106–115. Springer-Verlag, 2004.

[4] M. Pistore, P. Traverso, P. Bertoli, and A. Marconi. Automated synthesis of composite BPEL4WS Web services. In *Proceedings of 2005 IEEE International Conference on Web Services (ICWS 2005)*, pages 293–301, 2005.

[5] M. Sheshagiri, M. desJardins, and T. Finin. A planner for composing services described in DAML-S. In *Proceedings of the AAMAS Workshop on Web Services and Agent-based Engineering*, 2003.

[6] P. Traverso and M. Pistore. Automated composition of semantic web services into executable processes. In *Proceedings of 3rd International Semantic Web Conference (ISWC 2004)*, volume 3298 of Lecture Notes in Computer Science, pages 380–394. Springer-Verlag, 2004.

[7] R. Waldinger. Web agents cooperating deductively. In *Proceedings of FAABS 2000*, volume 1871 of Lecture Notes in Computer Science, pages 250–262. Springer-Verlag, 2000.

[8] Z. Manna and R. J. Waldinger. A deductive approach to program synthesis. *ACM Transactions on Programming Languages and Systems*, 2(1):90–121, 1980.

[9] M. Stickel, R. Waldinger, M. Lowry, T. Pressburger, and I. Underwood. Deductive composition of astronomical software from subroutine libraries. In *Proceedings of 12th International Conference on Automated Deduction (CADE 1994)*, volume 814 of Lecture Notes in Artificial Intelligence, pages 341–355. Springer-Verlag, 1994.

[10] S. McIlraith and T. C. Son. Adapting Golog for composition of Semantic Web services. In *Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002)*, pages 482–493. Morgan Kaufmann, 2002.

[11] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31(1-3):59–83, 1997.

[12] D. McDermott. Estimated-regression planning for interaction with Web services. In *Proceedings of the 6th International Conference on AI Planning and Scheduling*. AAAI Press, 2002.

[13] J. Rao, P. Küngas, and M. Matskin. Composition of semantic Web services using linear logic theorem proving. *Information Systems, Special Issue on the Semantic Web and Web Services*, 31(4-5):340–360, 2006.

[14] R. Hull and J. Su. Tools for composite Web services: A short overview. *SIGMOD Record*, 34(2):86–95, 2005.

[15] D. Berardi, D. Calvanese, G. de Giacomo, M. Lenzerini, and M. Mecella. Automatic composition of e-services that export their behavior. In *Proceedings of the First International Conference on Service-Oriented Computing (ICSOC 2003)*, volume 2910 of Lecture Notes in Computer Science, pages 43–58. Springer-Verlag, 2003.

[16] T. Bultan, X. Fu, R. Hull, and J. Su. Conversation specification: A new approach to design and analysis of e-service composition. In *Proceedings of 12th International World Wide Web Conference (WWW 2003)*, pages 403–410, 2003.

[17] S. V. Hashemian and F. Mavaddat. A graph-based approach to Web services composition. In *Proceedings of 2005 IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2005)*, pages 183–189. IEEE Computer Society, 2005.

[18] L. de Alfaro and T. A. Henzinger. Interface automata. In *Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering(ESEC 2001)*, pages 109–120. ACM Press, 2001.

[19] Information technology in public administration of Estonia. Yearbook 2006, Estonian Ministry of Economic Affairs and Communication, 2007.

[20] E. Tyugu. *Algorithms and Architectures of Artificial Intelligence*. IOS Press, 2007.

[21] S. Maslov. *Theory of Deductive Systems and Its Applications*. MIT Press, Cambridge, MA, 1987.

[22] G. Mints. *A Short Introduction to Intuitionistic Logic*. University Series in Mathematics. Springer, 2001.

[23] M. Matskin, R. Maigre, and E. Tyugu. Compositional logical semantics for business process languages. In *Proceedings of Second International Conference on Internet and Web Applications and Services (ICIW 2007)*. IEEE Computer Society, 2007.

[24] M. Matskin and E. Tyugu. Strategies of structural synthesis of programs and its extensions. *Computing and Informatics*, 20:1–25, 2001.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-59

Evolving Computer Game Playing via Human-Computer Interaction: Machine Learning Tools in the Knowledge Engineering Life-Cycle

Dimitris KALLES [a,b,1] , and Christos KALANTZIS [a,c]
[a] *Hellenic Open University, Patras, Greece*
[b] *Open University of Cyprus*
[c] *University of Patras*

Abstract. In this paper we review our work on the acquisition of game-playing capabilities by a computer, when the only source of knowledge comes from extended self-play and sparsely dispersed human (expert) play. We summarily present experiments that show how a reinforcement learning backbone coupled with neural networks for approximation can indeed serve as a mechanism of the acquisition of game playing skill and we derive game interestingness measures that are inexpensive and straightforward to compute, yet also capture the relative quality of the game playing engine. We draw direct analogues to classical genetic algorithms and we stress that evolutionary development should be coupled with more traditional, expert-designed paths. That way the learning computer is exposed to tutorial games without having to revert to domain knowledge, thus facilitating the knowledge engineering life-cycle.

Introduction

Biological aspects of human intelligence relate to how we unconsciously organize a massive set of brain neurons and their associated synapses. In that sense, any intelligence measurement (IQ) may be quite fair: it reflects whether some human has been able to adapt his (her) set of neurons as effectively as the IQ test would require. This fairness observation has to do with the fundamental fact that we are all born with roughly the same amount of brain tissue (neurons and synapses) and it is then the combination of genetic material (initialization and learning parameters) and the environment (input data) that lead each individual neuronal structure to evolve its own set of synapse weights.

Classical AI research in games has explicitly taken a different path in trying to mimic intelligence [1, 2]. Therein, model size is a key factor for success. Consider, for example, mini-max trees or alpha-beta pruning, where the look-ahead parameter allows us to employ increasing levels of sophistication that are reflected in the size of the model. So, taking chess as a classical example, the more difficult the level of playing, the

[1] Corresponding Author: Dimitris Kalles, dkalles@acm.org

larger the look-ahead that a computer employs to thwart the average human player's rather limited capability for exploring multiple alternative solutions.

A more human-oriented approach seems to be the training of a computer player. We have been using this option in our experimental research, mainly because it is inherently more interesting to investigate how to best capture human knowledge that is not available in the form of prescribed actions but, instead, is gradually demonstrated through interaction with an environment. This paper reviews our work in this area and shows how the automatic development of game-playing skills can be closely and measurably related to expert participation in game playing. It also shows why we think that game development warrants a priorities shift away from strict evolution towards the development of an integrated experimentation environment.

The paper is structured in four subsequent sections. Section 1 briefly presents our strategy game workbench. Section 2 reviews the key questions we raised and how we devised our experimentation to gain insight into answering them. In Section 3 we discuss the research options we are considering and, finally, we conclude by tying this research to non-evolutionary AI.

1. A Brief Background on a Strategy Game Workbench

The game is played on a square board of size *n,* by two players. Two square bases of size *a* are located on opposite board corners. The lower left base belongs to the white player and the upper right base belongs to the black player. At game kick-off each player possesses *β* pawns. The goal is to move a pawn into the opponent's base.

The base is considered as a single square, therefore every pawn of the base can move at one step to any of the adjacent to the base free squares. A pawn can move to an empty square that is either vertically or horizontally adjacent, provided that the maximum distance from its base is not decreased (so, backward moves are not allowed). Note that the distance from the base is measured as the maximum of the horizontal and the vertical distance from the base (and not as a sum of these quantities). A pawn that cannot move is lost (more than one pawn may be lost in one round). If some player runs out of pawns he loses.

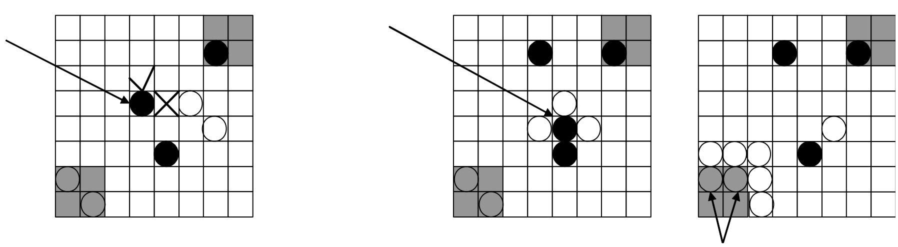


Figure 1. Exemplifying the rules of the game

The leftmost board in **Figure 1** demonstrates a legal and an illegal move (for the pawn pointed to by the arrow - the illegal move is due to the rule that does not allow decreasing the distance from the home base). The rightmost boards demonstrate the loss

of pawns (with arrows showing pawn casualties), where a “trapped” pawn automatically draws away from the game. As a by-product of this rule, when there is no free square next to the base, the rest of the pawns of the base are lost.

The *a priori* knowledge of the system consists of the rules only. To judge what the next move should be, we use reinforcement learning [3] to learn an optimal policy that will maximize the expected sum of rewards in a specific time, determining which action should be taken next given the current state of the environment. We approximate the value function on the game state space with neural networks [4], where the each next possible move and the current board configuration are fed as input and the network outputs a score that represents the expectation to win by making that move.

2. On Measuring the Effect of Human Involvement in Learning

Earlier experimentation [5] initially demonstrated that, when trained with self-playing games, both players had nearly equal opportunities to win and neither player enjoyed a pole position advantage. Follow-up research [6] furnished preliminary results that suggested a computer playing against itself would achieve weaker performance when compared to a computer playing against a human player.

The results of that earlier follow-up research demonstrated that even a limited amount of human involvement could deliver stronger performance of computer players that subsequently used the models developed by humans for their own play. By demonstrating that effect, they raised the all too obvious question of how could one judiciously apply human knowledge when training computer players, as opposed to just planning a periodical human interference. And last, they hinted that perhaps the answer to that question, namely the judicious involvement of human players, was the key to eventually develop a good playing machine.

To-date, while we have invested in several millions of games where automatic game-playing has been recorded and played-back, with different configurations of learning policy and neural network parameters and while we have not yet exhausted the alternatives for our experiments, we have far exceeded the number of games Tesauro [7] used for his legendary Neurogammon experiment and, nonetheless, we are yet nowhere near the level of a human player. Admittedly, we have not attempted to optimize the length of our training experiment and we may not have implemented the techniques in their full potential, as fellow researchers may have, but it seems that even the sophisticated arsenal of reinforcement learning and neural networks need a significant level of engineering attention so that they be fine-tuned to our application.

In this section we review our recent work on experimenting with varying levels of human involvement in training computer players. These experiments have to-date explored two different directions: one is that of experimenting with the parameters of the learning mechanisms (focusing on reward schemes) and the other is that of experimenting with the periodicity of human involvement and its effect based on the learning history prior to that involvement. The results have also helped us improve the resolution of our earlier findings [6].

2.1. On Tuning the Learning Mechanism

To detect improvement in automatic game playing, we have experimented [8] with different reward policies while constraining the moves of the human (training player) by navigating North, then Right, attempting to enter from the vertical edge of the black base (see **Figure 2**).

Experiments were organized in batches of 50,000 computer-vs.-computer (CC) games, carried out in 5 stages of 10,000 games each. For batches that have involved human-vs.-computer (HC) games, each CC stage is interleaved with a HC stage of 10 games. Thus, HC batches are 50,050 games long and, therein, a human is always the white player. Needless to say, at that early stage of our game we cannot lose, unless we explicitly wander and avoid entering the black base (as a matter of fact we have not observed any noticeable learning behavior in a series of experiments where we intentionally applied a *do-not-opt-to-win* strategy).

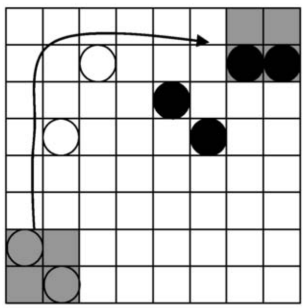


Figure 2. The path of a human player [8]

The alternatives for the rewards are shown in **Table 1**. Briefly describing them, the first one assigns some credit to states that involve a pawn directly neighbouring the enemy base. It also rewards capturing pawns by calculating the difference of pawn counts and by scaling that difference to a number in [-1,1]. The other two policies have been developed with a view towards simplification (by dropping the base adjacency credit) and towards better reward (penalty) scaling.

Table 1. Reward types [8]

	Win: 100, Loss: -100, and …
1	Pawn difference scaled in [-1,1] Win-at-next-move: 2, Loss -at-next-move: -2 (base adjacency credit)
2	Pawn difference scaled in [-1,1]
3	Pawn difference scaled in [-100,100]

The results of the corresponding three HC batches suggested that only the third policy delivered a clear improvement for the black player, as the number of moves required by the white CC player consistently increased, indicating that the black player

defended with increasing sophistication. The first two batches demonstrated a highly irregular behaviour; note that both are associated with a relatively non-fairly rewarded pawn advantage, and the first one even employs a very artificial mechanism to inject look-ahead knowledge in terms of neighbouring square sub-goals. These results were a clear indication that pawn advantage rewards should be commensurate with their expected impact in the game outcome; losing many pawns and not capturing any indicates that we are about to lose.

2.2. On Investigating Learning Acceleration

Follow-up experimentation clearly suggested that we should use shorter batches of 1,000 games each, to avoid state-space wandering. Sequences of 10,000 CC games in early game play were diluting the effect of knowledge instilled by earlier HC sessions.

We then moved to investigate [9] whether such acceleration could be employed in the involvement of the human player; after all a substantial cost to train a computer player is the time that must be expended by a human trainer. We also initially investigated how much prior self-learning may be useful. To do this we introduced the concept of tournament. A tournament between two batches B_X and B_Y is done in two steps where, first, the white player of the $B_X{}^{\text{th}}$ batch plays again the black player of the $B_Y{}^{\text{th}}$ batch; in the second step the inverse setting applies.

We report the tournament between batches B_1, B_2, B_3 and B_4. An extract of the experimental plan showing these batches is shown in **Figure 3**. Note that batches B_1 and B_2 are based on the same number of HC experiments, but batch B_2 is longer in terms of its CC experiments (since it is already based on X, a full CC batch). Batches B_3 and B_4 are similar but use just 1 game per HC stage. We also reprint in **Table 2** the results where, for each batch, we sum the differences of the number of games it has won against other batches minus the number of games it has lost. We also average, for each batch, the number of moves it made in winning games.

B_3		***X***		***B_4***	
Policy	1	**Policy**	N/A	**Policy**	1
Sequence	HC_1	**Sequence**	CC	**Sequence**	HC_1

B_1		***X***		***B_2***	
Policy	1	**Policy**	N/A	**Policy**	1
Sequence	HC	**Sequence**	CC	**Sequence**	HC

Figure 3. A short test for the acceleration hypothesis [9]

Table 2. Aggregate results in accelerated learning [9]

	Batch			
	B_1	B_2	B_3	B_4
Total number of games won	1006 (1)	712 (2)	-1226 (4)	-492 (3)
Average number of moves	284 (1)	286 (2)	438 (4)	375 (3)

Two important findings emerge. First, we observe a clear association between the ability to play fast and the ability to win games. Second, the two learning batches based on *really* sparse human involvement (1 game per HC batch) are abysmally low in performance. This suggests that there may be a lowest acceptable rate of human involvement, beyond which the effect of human training is effectively diluted.

2.3. Winners Usually Do Not Take Long to Win

To investigate the above hypothesis, we ran [9] an extended tournament between game configurations and we also experimented with a new navigation policy for the white player of the HC batches, beyond the trivial navigation strategy of "North, then Right". In the augmented policy, the white player (human) now attempts to cover all ground of the board in one HC stage, but with a different trajectory each time.

For example (see **Figure 4**), in the 5th game of a HC stage, the white player may start at the same cell as in the 2nd or the 4th game and will then proceed along the solid arrowed line. Only if the black player manages to effectively block access to its base will the white player be allowed to use a second pawn out of its base. In the 9th and 10th game of such a stage, the white player will move two pawns out of its base, and then proceed by moving in the central range of the board (shaded area).

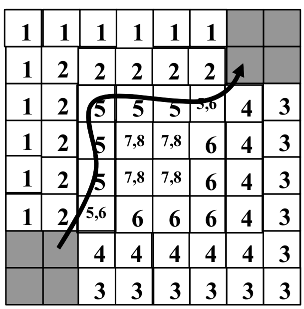


Figure 4. The augmented path of a human player [9]

The new experimental session provided us with the evidence that, wherever we observed a low average number of moves per game batch, therein was mostly probable that one of the tournament sides was comprehensively winning. In essence, the winning side had built both an effective strategy to overwhelm its opponent and an efficient one to do so fast. To substantiate this finding we devised two metrics to characterize each tournament game and we computed:

- The speed ratio as the ratio of the average number of moves per tournament leg.
- The advantage ratio as the ratio of wins by one player over the other per tournament leg.

The key finding is that small speed ratios generally deliver small advantage ratios. When speed ratios increase, so do advantage ratios. The only cases where this difference was less pronounced were the ones where the human had earlier navigated the whole board. The interleaved CC stages allowed for meaningful state-space exploration, based on a quite rich training content that increased subsequent game lengths (and

resulted in smaller speed ratios). To use a human learning analogy, a student who takes long to complete an assignment is, to an external observer who just observes the outcome (success or not) and the time spent on the assignment, equally likely to be a novice who takes long on an easy task or a sophisticated one who tackles a difficult problem (possibly, after some guidance by a tutor).

This also highlights a key danger in developing metrics for classifying the trade-off between speed and performance among competing approaches. Such metrics are only good at ranking competitors relative to each other and are not global metrics of game learning quality.

2.4. A Critical Appraisal of the Development Path

We definitely need more experiments if we are to train our computer players to a level comparable to that of a human player. Just experimenting with various aspects of the learning parameters is a challenge since we cannot directly attribute which part of the learning inefficiencies spotted in the long experimental runs of the previous sections may be due to some particular parameter setting.

The results we have obtained to-date clearly suggest that it is very important to find an efficient and effective way to achieve learning. We must optimize the use of expensive resources (the expert player) so that they are intelligently placed at critical points of the learning process, which will mostly be done automatically. Note that even though the number of HC games is relatively very small to the number of CC games, the impact of HC games can be clearly detected. Accurately measuring this impact is not straightforward, however. Therefore, it is of less importance to discuss how much to increase human involvement as opposed to gauging how to best spread a given amount of it.

Besides being a showcase of application of specific techniques and besides the goal of developing autonomous computer playing, investigating how to best utilise expert involvement has significant ramifications in the design of knowledge-based systems at large. The key observation is the identification of one archetypal design peculiarity of such systems: the elicitation of knowledge.

When confronted with strategy issues in an unknown field we, humans, tend to learn by trial and error. By confronting problems in a new domain we slowly develop measures of success and of measuring up the difficulty of the problem at hand. When on one's own, the selection of problems is a delicate exercise in resource management and can easily lead to exasperation. Much more effective is the employment of a tutor, if we can afford one.

Too coarse problems lead us to learn nothing while too fine problems focus us too much and render us unable to generalize if we are novices. Our research attempts to navigate the fine line between the sparseness and density of learning examples when the computer serves as the student [10, 11, 12] and the goal is to establish some examples of successfully tuning the "syllabus". Along that direction we expect that the length and the content of a training session will slowly become evident if we spend enough time even with very simple feedback.

A subtle point therein is that while the ratios are very similar, the average number of moves rises (probably) due to the richness in exploration possibilities incurred by the augmented path experiments (**Figure 4**). Such richness is necessary to improve the standard of the computer player by exposing it to learning opportunities. The automatic discovery of such opportunities is not trivial, however; finding one's way around nar-

rowly defined clusters of knowledge will probably lead to wandering [10]. In that sense, our recent attempts to provide playing examples by means of a mini-max player [13] also serve as a showcase of how to learn only based on observing an expert player particularly so as we can control look-ahead and then observe the impact on the learner.

3. Practical and Theoretical Considerations for Knowledge Elicitation

It is now interesting to review the development path to-date: we first experimented with "reasonable" learning parameters, then moved on to experiment with a "reasonable" duration of a learning batch and, following that, we experimented with the configuration of the "learning" material. While we have always tried to draw analogues to human tutoring, the only important and concrete "intelligence" milestone seems to be the experimental justification for the interestingness metrics (performance, speed) that we have eventually employed.

A reasonable development direction is to devise an experimentation engine that will attempt to calculate the best parameters for effective and efficient learning. This option has conceptual simplicity, technical appeal and has delivered some interesting results [14].

It is interesting to note that this direction is a close relative to the genetic algorithms paradigm; since it also evolves a player based on experience. Genetic programming *has* delivered results in game learning [15], yet we view that approach as inherently different to ours since it is also based on a description of games in terms of a vocabulary of features that constitute direct expert-derived knowledge about the particular game domain. Still, it is not a direction to be lightly dismissed, since a considerable line of research still involves the merging of low-level associative (similarity-based) search with higher-level (for example, spatial awareness) cognitive-based rules [16]. It is for this reason, we believe, that our experiments demonstrate measurable improvements when subjected to human tutoring. Though automatic playing has long been testified to deliver good results and still is a vibrant area, we emphasize human impact in a new game because we are interested precisely in exploring *disturbance* during learning.[2]

Interactive evolution is a promising direction. In such a course, one would ideally switch from focused human training to autonomous crawling between promising alternatives. But, as we have discovered during the preparation of this work, the interactivity requirements of the process of improving the computer player is very tightly linked to the availability of a computer-automated environment that supports this development. It is a must to strive to put the expert in the loop as efficiently as possible.

In terms of the experiments described above, we have noticed several features of an experimentation system that we have deemed indispensable if one views the project from the point of system efficiency. Such features range from being able to easily design an experimentation batch, direct its results to a specially designed database (to also facilitate reproducibility), automatically process the game statistics and observe correlations, link experimentation batches in terms of succession, while at the same time being able to pre-design a whole series of linked experiments with varying parameters of

[2] We use the term *disturbance* to denote the surprises that a human tutor can present to a learning machine.

duration and succession and then guide the human player to play a game according to that design.

As it seems, being able to provide a tool that captures the lifecycle of the development of an AI application is a strong contributor to the success of the take-up of that application. In that sense, we aim to primarily pursue the directions towards the automatic discovery of knowledge in game playing and secondarily towards equipping the computer with more detailed domain modelling [17] or with standard game-tree search techniques. One needs to improve interaction if one aims for interactive evolution.

Alongside the above applied research directions, one also needs the formulation of a language that allows the description of experimentation plans at least along the intuitively appealing notation employed in **Figure 2** and in **Figure 4**. The essence of moving towards that direction is that we need to be able to guide the evolution of the game engine along specific learning contexts, just how an experienced tutor does when guiding a student. Right now, this is only done by example of the expert tutor, who must be present, however. Note that such a development would help us spell out a moving strategy but says nothing about tactics. These are left to the human player to judge. There are several complications to consider:

- What would it mean to deviate a *little* from a prescribed path?
- What happens if we deploy more that one pawn, along the same prescribed path?
- What happens if we need to impose some level of coordination between pawns?

It is important to understand that having a language that allows us to specify such training scenarios leads us directly to contemplating a second evolutionary level, more akin to pure (symbolic) genetic programming.

4. Conclusions

We have reviewed experimental evidence that autonomous playing in a strategy board game is a domain where judicious human involvement can lead to measurable improvement. The timing and scope (extent) of that involvement can be seen to produce a relative richness in the ability of autonomous players to navigate faster in the state space of available game tactics. That such behaviour emerges is a strongly encouraging result, particularly so since it is not explicitly implemented in the credit scheme of the autonomous players. The price for the emergence of such behaviour is the (up to now) realization that human involvement cannot be simply dispersed throughout an experimental plan but must be carefully exercised. The analogue to human tutoring is very interesting and straightforward.

In our earlier work [8] we had suggested that significant applied research is required for the establishment of tools that will streamline the experimentation process and that such workflow-like tools to assist the human game designer (and, researcher) are more important than the autonomous management of this process. Still, we believe, that the establishment of objective and credible success (interestingness) measures will be important in both directions.

We view both approaches above as, in fact, sharing a key assumption of our approach: whichever way one looks at it, artificial intelligence may be central to game development but cannot exist in a vacuum; its impact can only be felt when it is well placed in an engineering context.

Acknowledgements

This paper has not been published elsewhere and has not been submitted for publication elsewhere. The paper shares some setting-the-context paragraphs with some referenced papers by the same author, with which it is neither identical nor similar. All game data have been recorded for potential examination and reproducibility tests. These data, along with the game code, are available on demand for academic research purposes.

References

1. A. Samuel (1959). "Some Studies in Machine Learning Using the Game of Checkers", *IBM Journal of Research and Development* 3, pp. 210-229.
2. C. Shannon (1950). "Programming a computer for playing chess", *Philosophical Magazine* 41(4), pp. 265-275.
3. R. Sutton (1988). "Learning to Predict by the Methods of Temporal Differences", *Machine Learning* 3(1), pp. 9-44.
4. R. Sutton, A. Barto (1998). *Reinforcement Learning - An Introduction*, MIT Press, Cambridge, MA.
5. D. Kalles, P. Kanellopoulos (2001). "On Verifying Game Design and Playing Strategies using Reinforcement Learning", *Proceedings of ACM Symposium on Applied Computing, special track on Artificial Intelligence and Computation Logic*, Las Vegas.
6. D. Kalles, E. Ntoutsi (2002). "Interactive Verification of Game Design and Playing Strategies", *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*, Washington D.C.
7. G. Tesauro (1995). "Temporal Difference Learning and TD-Gammon", *Communications of the ACM* 38(3), 58-68.
8. D. Kalles (2007). "Measuring Expert Impact on Learning how to Play a Board Game", *4th IFIP Conference on Artificial Intelligence Applications and Innovations,* Athens, Greece.
9. D. Kalles (2008). "Player co-modelling in a strategy board game: discovering how to play fast", Vol. 39, No. 1, pp. 1-18.
10. S.L. Epstein (1994). "Toward an Ideal Trainer", *Machine Learning*, Vol. 15, No. 3, pp. 251-277.
11. H.J. van den Herik, H.H.L.M. Donkersa, P.H.M. Spronck (2005). "Opponent Modelling and Commercial Games", *Proceedings of IEEE 2005 Symposium on Computational Intelligence and Games*, Essex University, Colchester, UK. pp 15-25.
12. E. Long (2006). *Game Theoretic and Machine Learning Techniques for Balancing Games*. M.Sc. Thesis, University of Saskatchewan, Canada.
13. D. Kalles, D. Kanellopoulos (2008). "A Minimax Tutor for Learning to Play a Board Game," *Proceedings of the AI in Games Workshop, 18th European Conference on Artificial Intelligence,* Patras, Greece.
14. I. Partalas, G. Tsoumakas, I. Katakis, I. Vlahavas (2006). "Ensembe Pruning Using Reinforcement Learning", *Proceedings of the 4th Panhellenic conference on Artificial Intelligence,* Heraklion, Greece, Springer LNCS 3955, pp. 301-310.
15. M. Sipper, Y. Azaria, A. Hauptman, Y. Shichel (2007). "Designing an evolutionary strategizing machine for game playing and beyond", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 3(4), pp. 583-593.
16. E. Lock, S.L. Epstein (2004). "Learning and Applying Competitive Strategies", *Proceedings of AAAI-04 (19th National conference on Artificial Intelligence),* San Jose, CA, pp. 354-359.
17. A. Junghanns, J. Schaeffer (2001). "Sokoban: Enhancing General Single-Agent Search Methods using Domain Knowledge", *Artificial Intelligence*, Vol. 129, pp. 219-251.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-69

Bridging Knowledge Gaps in Engineering Companies - The Case of Pipeline River Crossings in Greece

Fragiskos A. BATZIAS[a] and Philip-Mark P. SPANIDIS[b,1]
[a]*Department of Industrial Management and Technology, University of Piraeus, Greece*
[b]*ASPROFOS Engineering SA, Greece*

Abstract. The introduction of Knowledge Management (KM) processes is suggested herein for bridging knowledge gaps observed during long-term and high complexity engineering projects, like the Natural Gas Project (NGP) of Greece, with a view to deploying a dedicated KM system in engineering companies. The model of three essential knowledge management processes (acquisition, maintenance and sharing) is demonstrated, applying the IDEF0 method of functional analysis and design. The functionality of the introduced KM processes has been proved by a case study referring to a pipeline river crossing sub-project.

Keywords. Business process modeling, engineering company, knowledge gap, natural gas, project management, technology transfer.

Introduction

Long-term and large-scale engineering projects present a variety of dissimilarities on technical concept, time or resource constraints. The knowledge of each project is neither standardized, nor easy to be formalized. However, as shown in the Natural Gas Project of Greece (NGP) [1], the role of knowledge is critical in project success and relates to philosophy, strategies and tools the engineering companies are using to manage corporate knowledge.

In literature, two types of knowledge are discussed: tacit and explicit. Tacit is personal, context specific and not easily transferable, whilst explicit is formal, codified, transferable and storable in (digital or conventional) archives [2,3]. What is important for an engineering company is the conversion of tacit knowledge to explicit and vice-versa. By this conversion, the performance of technical work is achieved, the corporate knowledge is assimilated among personnel and the company becomes a learning organization with strategic advantages against its competitors [4]. In small projects, knowledge is quite identifiable and controllable and can be managed by small and flexible expert groups. Personal capability, know-how and insight, is converted and embodied into studies including technologically advanced information. In multidisciplinary projects, on the other hand, knowledge conversion tends to be difficult and complex. Heterogeneous bodies of tacit and/or explicit knowledge originated from numerous stakeholders have to be systematically diffused within social networks (task forces, joint ventures, project teams) and then successfully converted in technical documents useful for contractors or material suppliers [1]. Under circumstances of technical complexity and social interaction, *knowledge gaps* are

[1] Corresponding Author: Email: fbatzi@unipi.gr; pspani@asprofos.gr

observed, especially when the technical organizations do not employ methods and tools for beneficial management of knowledge [1]. Thus, the most complex and the longest the project are, the highest the probability for observing knowledge gaps.

In order to manage knowledge acquiring or accumulating through projects or training, organizations make use of various internal processes applied to the pure technical work. These processes are formulated by means of business process models, the philosophy of which presents interesting perspectives from the Knowledge Management (KM) viewpoint. Being designed to deliver services and products to customers, business processes are meant as functions of structured and interrelated set of activities, transforming inputs into specified outputs [5,6]. In literature a lot of attention has been paid on the contribution of business processes in the development of corporate knowledge into organizations and projects [7-9]. Knowledge is a key entity of the whole production system that can in a systematic and controllable manner be acquired, stored, reused or updated through KM processes. The knowledge processes constitute the backbone of KM systems by which productivity, services quality and assimilation of knowledge body by company's personnel is succeeded.

Reconsidering the NGP and the knowledge gaps that occurred during its implementation [1], it is assumed that *lack of knowledge* arises when a company does not avail tacit/explicit knowledge in the required appropriateness level, content and quality or cannot transfer knowledge that is originated from various sources. *Knowledge loss* occurs when company's knowledge is missing or inaccessible or wrongly recorded. *Knowledge misuse* reflects combination of previous gaps and refers to misconceptions, misperceptions or implications among company's personnel generated due to poorly delivered or undeveloped corporate knowledge as well as from weaknesses of company's training system.

This paper attempts to provide a solution for bridging knowledge gaps observed in long-term, high complexity and large-scale engineering projects, like the NGP [1], by introducing a business process model architecture, consisted of three interrelated KM processes: (a) a *knowledge acquisition (KAQ)* process for dealing with *lack of knowledge,* aiding the efficient management of knowledge bodies that the company acquires during projects or by other non-project sources (e.g. training or Research and Development-R&D); (b) a *knowledge maintenance (KMT)* process for preventing *knowledge loss,* based on the advanced, automated and intelligent storing of knowledge portions developed into the company's organizational barriers; (c) a *knowledge sharing (KSR)* process for precluding *knowledge misuse,* aiming at familiarizing the company's personnel with purposes, content and applicability of corporate knowledge.

1. The Knowledge Management Conceptual Model

1.1. Assumptions and KM Model Description

Assumption-1: The corporate knowledge can be stored into digital a *Knowledge Base (KB)*. The KB would be a real-time system allowing company's personnel to retrieve information on past projects and from associated engineering, technological and scientific fields.

Assumption-2: The engineering company will maintain a *Knowledge Engineering* department for managing corporate knowledge. In small companies, this work can be assigned to experts, each undertaking a specific domain of knowledge acting as knowledge engineering coordinator.

The KM conceptual model is presented in Fig.1. The knowledge processes (KAQ, KMT, KSR) run synergistically with engineering production when requirements of a certain project (scope, technical data, feasibility reports, etc.) are addressed. The duty of the knowledge engineering department is to give functional substance to knowledge processes. The KB plays the role of organization's selective memory in an easily retrievable form, as extracted from previous projects (explicit knowledge), valuable for managers and engineers. R&D/Training is a process established to facilitate upgrading of technological know-how, insofar personnel after apprenticeships becomes transferor/facilitator of new practices/methods (tacit knowledge). R&D/Training also enables updates on innovative achievements in conventional or digital reports (explicit knowledge). Finally, know-how and technology transfer (KTT) represents a process evolving social networks (e.g. collaborations, joint ventures, etc.), when knowledge gaps are observed.

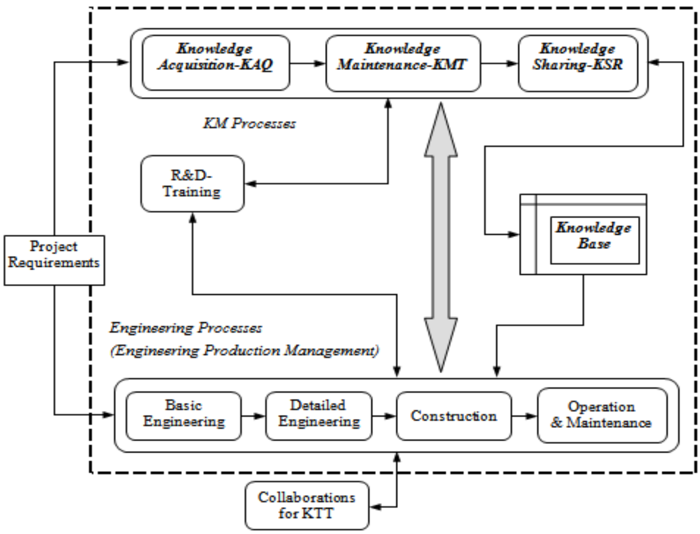


Figure 1. The Conceptual KM Process Model.

1.2. The Knowledge Base

The KB comprises the core element of the KM processes functionality, containing the following subsystems:

(a) *Engineering and Management Guide*: codes, standards, handbooks, specifications for e.g. LNG, pipelines and compressors, methods of linear geophysical research, crossings rehabilitation and quality/safety/inspection technologies; data from pipelines projects for dimensioning of NGP units by means of similarity theory and practice corresponding to software packages performing computer aided dimensional analysis [10]; managerial information (know-what/who/how, information for suppliers, subcontractors, consultants, institutions and KTT costs), project/engineering/construction management information.

(b) *Empirical Guide*: empirical rules of thumb, lessons learned from oil/gas exploitation (drilling, welling, risers installation), troubleshooting of failed pipe systems operation, welding repairing, expertise rules for corrosion recognition (by means e.g. of fault tree analysis-FTA) or prevention [11] and best practices applied on oil and gas systems installation, operation and maintenance.

(c) *Literature Subsystem*: papers from scientific literature, e.g. hydrocarbon reservoir fluid mechanics, optimization of pipe networking, geophysical models, risk analyses of oil and gas industry plants and facilities, gas systems metering/regulating, control and remote operation architectures, innovative equipment and materials and finally, scientifically accredited reports useful for technical applications, justification of engineering work and company's R&D.
(d) *Energy Records*: historical data collected from the domestic/worldwide energy industry/economy (time series of fuels price, energy market reviews, techno-economic analyses, etc.) presented in various formats to facilitate forecasting of natural gas consumption and supply; links with domestic/European legislation to encompass energy policies or investment initiations.
(e) *Ontological Vocabulary*: project terminology, expressions and semantics of scientific or technical literature, in order to avoid misunderstanding between delegates representing different organizations (participating within the same consortium that has undertaken a project) and possible different schools of thought [12]; these vocabularies should be continually updated/enriched by means of an intelligent agent acting as an interface with external data/information/KBs [13].

2. Case Study

The KM model proposed herein has been implemented retrospectively in a project of natural gas pipeline crossing at the lagoon of river Nestos, using the IDEF0 method for modeling the knowledge processes. Emphasis has been given to the detailed engineering phase in order to show how (i) construction methods are evaluated making know-how elicitation from an experts group through a multi-criteria analysis (MCA), (ii) technical experience (meta-knowledge) can be transformed to maintainable knowledge entities and (iii) knowledge can be shared across an engineering company's departments.

2.1. Basics of River Crossings Engineering

The most applicable method for underground crossing of pipelines is the horizontal directional drilling (HDD), recognized as an environmentally friendly method although it draws high application cost and performance risk [14]. Other methods are: Bottom Pulling (BPM: the pipe is covered by concrete coating and afterwards pulled into the river in a trench opened across to the river bed), Open Cut (OCM: the pipe is pulled into a trench opened across to the river bed and stabilized with concrete saddles) and Pipe Laying (PLM, applicable in small waterways: the pipe is armored by concrete coating and afterwards is laid into the river bed). The phases of river crossing projects are as follows:
(a) *Basic Engineering*: Definition of pipe alignment, identification of the environmental protection zones and gathering of geological background data necessary for the crossing method evaluation, engineering and construction.
(b) *Detailed Engineering*: Geotechnical research, definition of alternative construction methods, Multi-Criteria Analysis (MCA) for selection of the adequate construction method, design of crossing geometry, site development requirements for the work execution, method cost analysis, KTT subcontracting for the geotechnical research (if required) or engineering studies and preparation of tender documentation.
(c) *Construction*: Pipeline installation according to the selected construction method.

2.2. The IDEF0 method

The IDEF0 method, proven suitable in functional analysis and modeling of diverse industrial, manufacturing and business processes [15,16] has been selected by the authors for the design of the proposed knowledge processes. According to IDEF0 any process is represented by a box, in the boundaries of which several arrows are coming or leaving as input (I) or output (O) entities representing data, information or results of other processes. Arrows coming or leaving from the top or the bottom of the box represent respectively control factors (C) influencing the process functionality and mechanisms (M) necessary for process functionality. The overall abbreviation of the IDEF0 model is referred as IOCM. In a detailed level of analysis, each parental process may be dived to other internal sub-functions (tasks/sub-processes) and so forth. Fig. 2 illustrates the internal tasks and the associated IOCM entities for the pipeline river crossing case study.

2.3. The KM Processes Analysis

2.3.1. KAQ

This process includes three fundamental tasks: (a) collection and classification per knowledge domain of the technical data for the crossing, as they are outlined from basic engineering phase; (b) collection and preliminary evaluation of any information coming from R&D or training in fields of underground construction technologies; (iii) Multiple-Criteria Analysis (MCA) for evaluation of the construction method to be engineered and applied for the certain crossing with respect to a set of techno-economical and environmental criteria. The MCA analysis is the most critical function, through which the tacit knowledge of a selected experts' group is elicited and then converted to explicit by a two round DELPHI application method [17]. Based on data gathered from questionnaires, a quantitative analysis is performed applying a neuro-fuzzy approximation algorithm to determine ranking of alternative crossing methods and sensitivity analysis of each criterion [17]. Figure 3 shows the MCA summary table that depicts the output of the applied KAQ process. For assuring objectivity, the experts selected do not include the personnel involved in Nestos crossing.

2.3.2. KMT

This process includes three fundamental tasks: (i) organization and codification of knowledge topics embodied into detailed engineering or post-KAQ documentation (e.g. MCA report and methodological aspects), (ii) establishment of an expert system demonstrating probable engineering or construction failures that should be derived in this type of crossings and (iii) codification and archiving of any report or lesson learned that should be included in post-engineering or post-construction phases (meta-knowledge). In the case study presented herein, the HDD failures have been analyzed by means of FTA, a broadly adopted technique for knowledge representation [18]. The authors demonstrate how the HDD experience (tacit knowledge) is formulated and hence codified in a tree-structure (explicit knowledge) allowing storage or accessibility in company's KB system. An FTA extract is depicted in Fig.4a as a part of an implemented expert system that also includes records of pipeline failures like corrosion and routing defects. This expert system provides information useful in fields of pipe protection and coating, microbiologically induced corrosion (MIC) of pipelines at alluvial-waterway settlements (Fig.4b), hydrogen induced stress corrosion cracking and cathodic protection.

(a)

Process: KAQ [P(1)]		
Inputs	I(1,1)	: Basic Engineering data (pipe diameter/pressure, geo-environmental data, legislation, crossing length)
	I(1,2)	: Meta-Knowledge updates (cases studies, lessons from previous crossing projects)
	I(1,3)	: Determination and statement of crossing construction methods
Outputs	O(1,1)	: Selected R&D-Training reports (in subsurface/rock mechanics, micro-tunneling/drilling technologies)
	O(1,2)	: KTT requirements for Detailed Engineering elaboration
	O(1,3)	: Company's know-how availability/reusability report
	O(1,4)	: MCA report and methodological statements (evaluation of most adequate contruction method)
Controllers	C(1,1)	: Compnay's availability of experts in river crossing engineering/construction technologies
	C(1,2)	: Company's R&D-Training budget
Mechanisms	M(1,1)	: Knowledge Engineering Department
	M(1,2)	: Project Management Department
	M(1,3)	: Engineering Departments
Process: KMT [P(2)]		
Inputs	I(2,1)	: MCA reports and methodological statements [I(2,1)=O(1,4)]
	I(2,2)	: Meta-Knolwedge updates [I(2,2)=I(1,2)]
	I(2,3)	: Selected R&D-Training reports [I(2,3)=O(1,1)]
	I(2,4)	: Selected technical documentation (geotechnical/mechanical studies, project management data, etc.)
Outputs	O(2,1)	: Information for organization, engineering and implementation of gas pipeline river crossings
	O(2,2)	: Link with Expert System established for waterway crossing engineering failure analysis
	O(2,3)	: Post-engineering KTT qualification reports
	O(2,4)	: Project Vocabulary
	O(2,5)	: Crossing Engineering Literature Updates
Controllers	C(2,1)	: KB characteristics (storage capacity and capabilities of data processing architecture)
	C(2,2)	: Company's experts availability on intelligent/knolwedge based systems design/implementation)
	C(2,3)	: Company's budget for infromation systems development
Mechanisms	M(2,1)	: Knowledge Engineering Department
	M(2,2)	: Information Technologies Department
	M(2,3)	: Document Control Department
	M(2,4)	: Engineering Departments
Process: KSR [P(3)]		
Inputs	I(3,1)	: Selected technical documentation (geotechnical/mechanical studies, proj. management data, etc.)
	I(3,2)	: Post-engineering KTT qualification rep [I(3,2)=O(2,3)]
	I(3,3)	: Link with Expert System established for waterway crossing engineering failure analysis [I(3,3)=O(2,2)]
Outputs	O(3,1)	: Intra-organizational/regular reporting and updating
	O(3,2)	: Meta-knolwedge updates [O(3,2)=I(2,2)=I(1,2)]
Controllers	C(3,1)	: Company's users' interfaces and networking facilities
Mechanisms	M(3,1)	: Knowledge Engineering Department
	M(3,2)	: Information Technologies Department
	M(3,3)	: Document Control Department

(b)

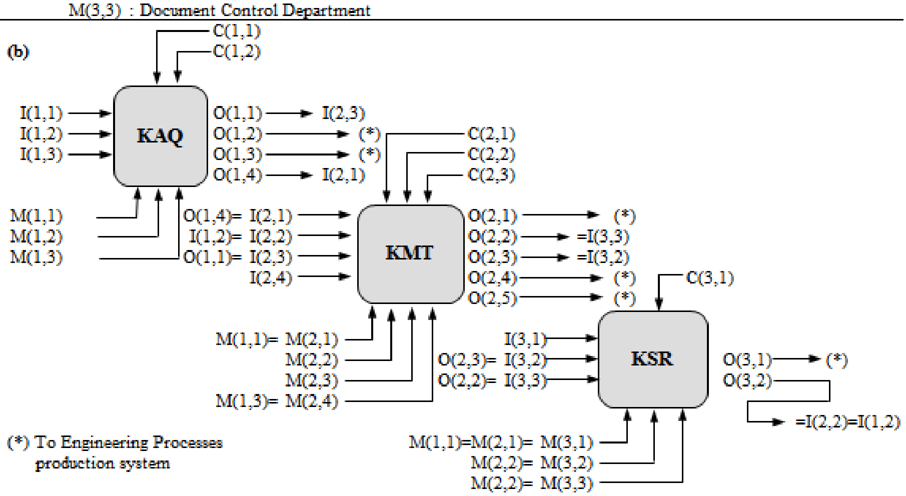


Figure 2. (a) IOCM entities of KM process model proposed for Pipeline River. (b) KM Processes System Design.

Project:	Pipeline River Crossing at Nestos Wetland, Thrace, Greece
MCA subject:	Evaluation of construction technology adequate for underground Gas Pipeline installation
Knowledge elicitation method:	2 round DELPHI (use of structured questionnaires) 1st round: ranking of evaluation criteria-2nd round: evaluation of construction methods
Evaluation criteria:	(f1): Construction & Maintenance Cost, (f2): Cost of Geotechnical Research (f3): Disturbance of Nestos Ecosystem, (f4): Pipeline Safety (f5):Know-How availability, (f6): Perspectives for Technology Transfer
Alternative Technologies:	(T1): Horizontal Directional Drilling-HDD, (T2): Bottom Pulling Method-BPM, (T3): Open Cut Method-OCM, (T4) Pipe Laying Method-PLM
Experts Qualifications:	(1) Experience: 10 years minimum in large scale pipeline projects (2) Scientific Skills: MSc/PhD post-graduate level of education (3) Position of duties: Engineering or Project Manager/Coordinator
Number of Experts:	Six (6)
Multicriteria Problem:	Max{f1(a),...f6(a) | a=T1, T2, T3 and T4}, f1, f2,...f6: the evaluation criteria
Outranking Method:	PROMETHEE-II
Generalized Criterion:	Piecewise Preference Linear Function P=H(d), 0< d < 1 d: difference of evaluation between two alternatives Tk and Ts | k, s =1, 2, 3, 4
Parameters of H(d):	p: preference threshold, q: indifference threshold
Decision Matrix and Criteria Weights:	Application of L-R triangular fuzzy numbers
Mathematical criterion:	Low-Sensitivity-High Robustness
Partial Ranking of Alternatives (PRA)	1st solution: High preferability Level q=0.5 and p=1.0 T1 → T2 → T3 → T4 The arrow means "better than" 2st solution: Mean preferability level q=1.5 and p=3.0 T1 → T2 → T4 → T3
Method Selected:	T1: HDD is the most robust alternative method
Sensitivity Analysis for each Criterion (SAC):	q=0.5 p=1.0 q=1.5 p=3.0

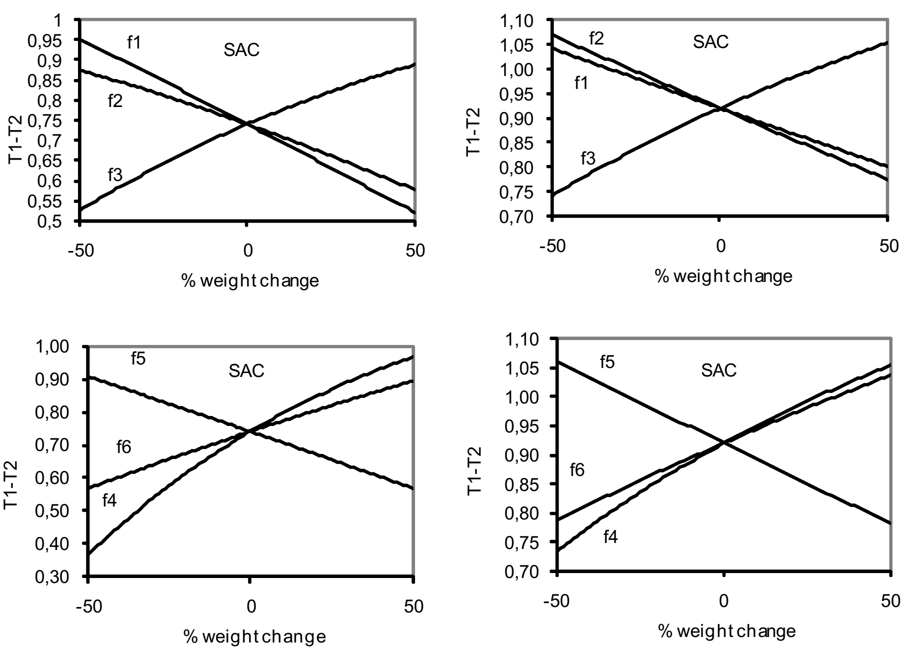
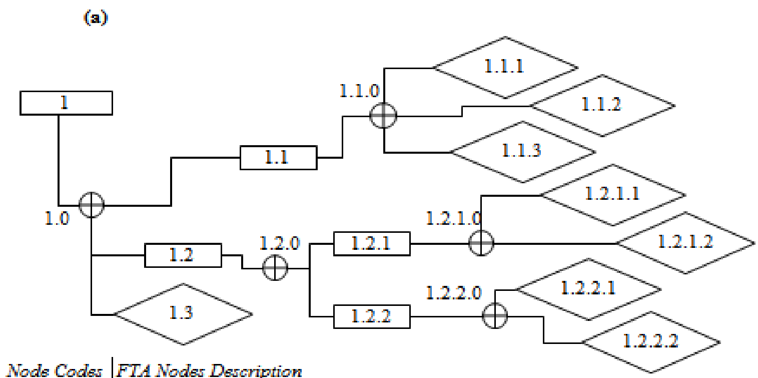
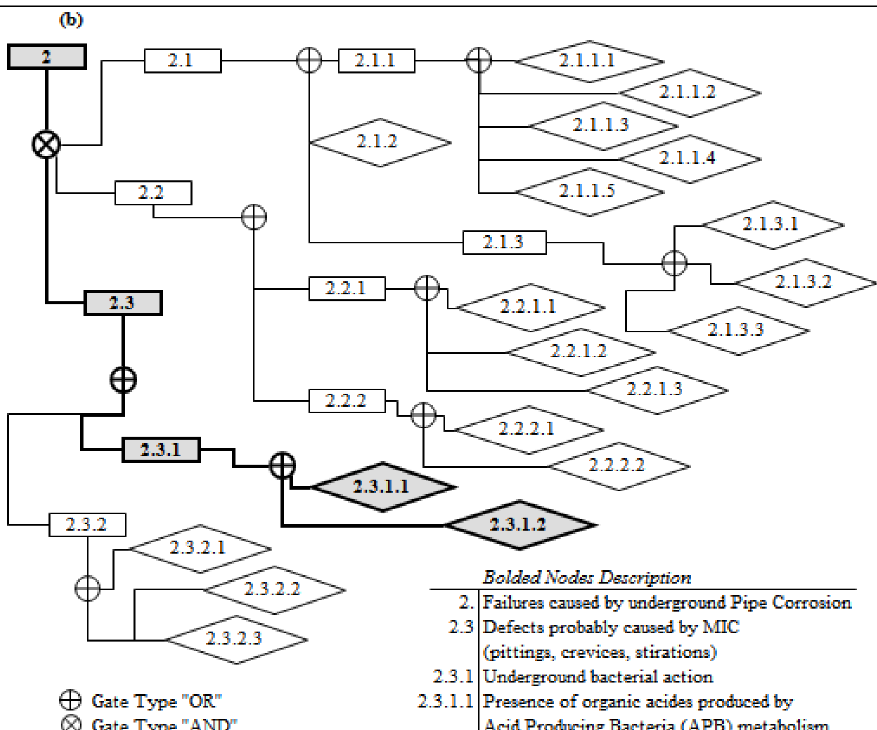


Figure 3. Summary Report of MCA task of KAQ Process.



Node Codes	FTA Nodes Description
1	HDD Engineering Failures
1.1	Misalignment of drilling hole
1.2	Obstruction of drilling hole opening
1.3	Groundwater flactuations due to natural/man-made interventions
1.1.1	Poor calibration/setting of underground guidance equipment
1.1.2	Inadvertent monitoring/execution of the subsurface work
1.1.3	Insufficient experience of drilling operators
1.2.1	Presence of rock formations
1.2.2	Presence of grain materials
1.2.1.1	Presence of cohesive rocks
1.2.1.2	Presence of fructured/weathered rocks
1.2.2.1	Presence of gravels/cobbles less than 50-100 mm
1.2.2.2	Presence of boulders/gobbles larger than 100 mm



	Bolded Nodes Description
2.	Failures caused by underground Pipe Corrosion
2.3	Defects probably caused by MIC (pittings, crevices, stirations)
2.3.1	Underground bacterial action
2.3.1.1	Presence of organic acides produced by Acid Producing Bacteria (APB) metabolism
2.3.1.2	Presence of FeS or $Fe(OH)_2$ due to Sulphide Reducing Bacteria (SRB) metabolism

Figure 4. (a) HDD Failures (FTA knowledge representation structure). (b) Pipeline MIC Analysis at Alluvial-Waterway formations (bolded path entities) – Extract from an Expert System.

2.3.3. KSR

This process deals with the systematic delivery of updates and reports to company's departments of knowledge acquired through engineering activities, as well as the meta-knowledge aspects cumulated when portions of an undertaken project is progressively completed. Fig. 5 illustrates a knowledge matrix sharing by means of an intra-organizational delivery model of reports and documents (selective tabulation) towards an HDD project.

	Reports & Documents for Intra-Organizational Delivery	Departments of a typical Engineering Company													
		Mechanical Process	Civil & Geotechnical	Surveying (ROW/Routing)	Environmantal & Permits	Instrumentation & Control	Electrical Engineering	Piping & Stress Analysis	Material Technology	Machinery & Pumping	Project Managemenet	Quality Management	Construction Suipervision	Inspection	Document Control
1	Multi-Criteria Analysis Report	L	M	H	H	_	_	_	_	_	H	M	_	_	(*)
2	Cause Effect diagrams of HDD Failure Analysis (based on FTA results)	L	L	L	M	_	_	M	H	L	M	M	H	H	(*)
3	List of critical engineering studies for HDD construction projects	L	H	M	H	L	L	H	H	L	H	M	M	_	(*)
4	Post-Engineering and Post-Construction lessons learned	L	H	H	H	L	L	H	M	1	H	M	H	M	(*)
5	Updates on HDD crossing technologies/innovations	_	H	H	M	L	_	L	M		M	_	M	_	(*)
6	Quality and Safety Issues for HDD engineering/perfromance	L	H	L	L	_	_	_	_	_	H	M	H	M	(*)
7	List of HDD consultants & subcontractors (for KTT	_	H	L	_	L	_	L	_	_	H	_	M	M	(*)
8	Scope of subcontracted works (KTT technical content)	L	H	M	L	L	L	M	L	L	H	_	L	_	(*)
9	HDD Project Vocabulary (Semantics)	L	H	H	H	L	L	H	M	_	H	M	H	L	(*)
10	Literature Updates (on line scientific libraries, oil & gas institutes, etc)	L	H	H	M	L	L	M	M	L	L	L	L	_	(*)
11	e-knowledge links for Underground Technologies (web links/portals)	M	H	H	H	L	L	M	M	_	L	L	M	M	(*)
12	Project Issues (samples of Schedules, Work Breakdown Analyses, Contracts)	_	M	M	L	L	_	L	L	_	H	M	H	M	(*)

Indicators of significance :H=high, M=medium, L=Low, '-'=None (*): For archiving purposes

Figure 5. Knowledge Sharing (KSR) Matrix.

3. Discussion and Concluding Remarks

The KM model proposed herein is beneficial in mitigating knowledge gaps, reducing knowledge acquisition costs, and improving know-how manageability. The IDEF0 tool offers the advantage of simplicity in IOCM entities representation and modeling and adaptability to computational methods as they have been used for industrial/business processes simulation. Methods/tools of knowledge elicitation or representation (MCA, FTA) are inherently part of the KM processes, facilitating the transformation of tacit knowledge or collective experience to codified entities that are maintainable and, therefore, reusable by company's personnel at any time. The methodology proposed herein, expands the typical project management framework with the introduction of knowledge (pertinent to/acquired from the project) as a multidisciplinary functional element, interactively associated to the engineering work production, in order to increase the company's capabilities and competitive advantage.

The business processes concept adopted by the authors can be fitted within any engineering company, considering that (i) personnel, being accustomed to working systematically in engineering processes, will adapt promptly to these new processes,

(ii) the company's departmentalization enables the design of knowledge processes in association with the work breakdown of a given project, and (iii) being responded to the theoretical model 'input-process-output', the processes can be simulated by data processing computational methods. The adaptability of processes to information technologies is '*sine-qua non'* issue for company's shifting to the computer-aided against the traditional way of working. The necessity of knowledge explicitness is widely promoted by means of intelligent/information system technologies especially in fields of multidisciplinary project management and technical consulting.

Acknowledgements

Financial support provided by the Research Centre of the Piraeus University is kindly acknowledged.

References

[1] F. Batzias, P.-M.P Spanidis, Identifying Knowledge Gaps in Engineering Companies - The Case of the Natural Gas Project in Greece, *Proc. 18th Intern. Congr. Chem. Proc. Eng. (CHISA), Prague, Czech Republic* (2008), in press.

[2] I. Nonaka, P. Byosiere, C. Borucki, N. Konno, Organizational knowledge creation theory: A first comprehensive test, *International Business Review* **3**(4) (1994), 337-351.

[3] I. Nonaka, P. Reinmoeller, D. Senoo, Management focus: The "ART" of knowledge: Systems on capitalize on market knowledge, *European Management Journal* **16**(6) (1998), 673-684.

[4] A. Berry, K. Oakley, Consultancies: agents of organizational development-Part 1, *Leadership and Organization Development Journal* **15**(1) (1994),13-21.

[5] T.H. Davenport, *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press, Boston, MA, 1993.

[6] D.A. Garvin, Leveraging Processes for Strategic Advantage, *Harvard Business Review* **73**(1995), 76-90.

[7] B. Kalpic, P. Bernus, Business process modeling in industry–The powerful tool in enterprise management, *Computers in Industry* **47**(3)(2002),2 99-318.

[8] R. Hall, P. Andriani, Managing knowledge associated with innovation, *Journal of Business Research* **56**(2) (2003), 145-152.

[9] J. Jung, I. Choi, M. Song, An integration architecture for knowledge management systems and business process management, *Computers in Industry* **58**(1) (2007), 21-34.

[10] F.A. Batzias, A.S. Kakos, N.P. Nikolaou, Computer–aided dimensional analysis for knowledge management in chemical engineering processes, *Journal of Computational Methods in Science and Engineering* **5** (2005), 47-59.

[11] P.J. Graham-Jones, B.G. Mellor, Expert and knowledge-based systems in failure analysis, *Engineering Failure Analysis* **2** (1995), 137-149.

[12] F.A. Batzias, P. Spanidis, An ontological approach to knowledge management in engineering companies of the energy sector, *Sustainable World, International Series Energy and Environment*, WIT Press Computational Mechanics Inc. Boston Massachusetts, USA, **7** (2003), 349-359.

[13] F.A. Batzias, E.C. Marcoulaki, Restructuring the keywords interface to enhance computer aided process engineering, *Computer Aided Chemical Engineering* **10** (2002), 829-834.

[14] E.N. Allouch, S.T. Ariaratnam, J.S. Lueke, Horizontal directional drilling: Profile of an emerging industry, *ASCE, Journal of Construction Engineering Management* **126** (2000), 68-76.

[15] A. Kusiak, T.N. Larson, W. Juite, Reengineering of design and manufacturing processes, *Computers and Industrial Engineering* **26** (1994), 521-536.

[16] K.O. Yusuf, N.J. Smith, Modeling business processes in steel fabrication, *International Journal of Project Management* **14**(6) (1996), 367-371.

[17] A.F. Batzias, F.A. Batzias, Computational and experimental process control by combining neuro-fuzzy approximation with multi-criteria optimization, *11th Int. Conf. Computational Methods and Experimental Measurements (CMEM)*, WIT Press (2003), 107-118.

[18] I.H. Witten, E. Frank, *Data Mining-Practical Machine Learning Tools and Techniques*, 2nd Edition, Elsevier, 2005.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-79

Implementation Issues of a Knowledge-based Geographical Information System

Katerina KABASSI[1], Eleni CHAROU[1,2], Aristotelis MARTINIS[1]

[1]*Department of Ecology and the Environment, TEI of the Ionian Islands, 2 Kalvou Sq, 29100, Zakynthos, Greece*

[2]*Institute of Informatics and Telecommunications NCSR DEMOKRITOS P.Grigoriou , 153 10 Ag. Paraskevi, Greece*

E-mail: kkabassi@teiion.gr; exarou@iit.demokritos.gr; amartinis@teiion.gr

Abstract. The research presented in this paper introduces a user context approach for the implementation of an adaptive Geographical Information System (GIS). The main focus of the paper is on presenting some implementation issues about the data used and their evaluation with respect to their suitability for the user interacting with the system. For the evaluation of the geographical information, the system uses a simple decision making model and selects the one that seems more appropriate for a user. In this way, the GIS has the ability of adapting its interaction to each user and make interaction more user friendly.

Keywords: Geographical Information System, Knowledge Based Systems, Human Computer Interaction

Introduction

The Geographical Information (GI) industry is a specialized component of the broader information technology sector and has scientific and technical links to many other disciplines such as environmental science, engineering, computer science, health delivery, planning and resource management. Geographical information is fundamental to our everyday lives. Satellite images bring daily weather reports; global positioning systems monitor the location of thousands of trucks and taxis; real estate sales use geographic information systems; and maps of all kinds are produced, displayed and analyzed using the Geographical Information technology. However, Geographical Information Systems (GISs) are usually targeted to scientists for the environment and other users who are not specialists find them confusing. A remedy to this problem is the development of systems with an ability to adapt their behaviour to the interests and other features of individual users and groups of users (Virvou 2001).

Given the popularity of geographical data and the variety of users groups dealing with this data it is desirable to develop Geographical Information Systems adaptable to

the users needs and skills. Indeed, lately there is an increasing interest for personalized GIS for making recommendations and for this purpose several techniques have been proposed (Malpica et al. 2007, Choi 2007).

In view of the above we have developed ADAPTIGIS (Kabassi et al. 2006), a knowledge-based GIS that can adapt its interaction to each individual user. In order to evaluate different geographical information, the system uses a simple decision making model. The information that is rated highest by the decision making model is selected to be presented by the system.

1. Data used and GIS implementation

A Geographical Information system has been developed for the Zakynthos island in Greece. A number of topographic features were digitized from Topographic Maps of the Geographic Service of the Army (scale 1:50.000). Topographical data include the coastline, the main and secondary road network, meteorological stations and village polygons (outline of village limits). A similar procedure was followed in the digitization of the geological maps of the Institute of Geological & Mineral Exploration IGME (scale 1: 50.000) and soil maps (land use and land capability for forestry) of the Ministry of Agriculture (scale 1: 20.000).

Table 1: Pre-Processing/Image Enhancement/Classification

Technique	Result
Georeferencing	Image map output in Hellenic Projection System of 1987
Color Composites	Best combinations for Landsat data are achieved using bands TM 1,3 (or 4) and 5 (or 7) as well as real color composites.
Intensity Hue Saturation HIS Images	Images are enhanced while shadow is suppressed.
Unsupervised classification using Self Organizing Maps.	Interpretation of spectral characteristics of images. Easy discrimination of land cover classes.
Automatic conversion of raster to vector data.	Map output. Inform the GIS database with the output vector data
Collection / input / coding, Storage/ Management, Retrieval, Processing / analysis, Presentation / Display, & Map making	Creation of a relational database of the collected data, map making. Evaluation of temporal changes, map updating.

Geologic layers (vector) containing the hydrological network, lithological unit boundaries, tectonics (faulting and bedding system) were created. Following the digitization of the maps, georeferencing of them was performed, by choosing specific Ground Control Points (GCPs) in the corresponding maps and the digitized coastline.

Accurate mapping of the most important sites of cultural and/or natural heritage, as well as mountainous footpaths of Zakynthos was carried out using a GPS Thales. For each path a description of the type of the path, the terrain involved, experience needed, estimated time required and a classification of the paths according to the difficulty was attempted.

Two Landsat 7 Enhanced Thematic Mapper Plus (ETM+) scenes have been used, with acquisition dates 28/07/1999 and 15/08/2000, respectively. Various image processing and vector GIS techniques have been applied for the analysis of the satellite imagery (Table 1). Some results are presented in Figure 1.

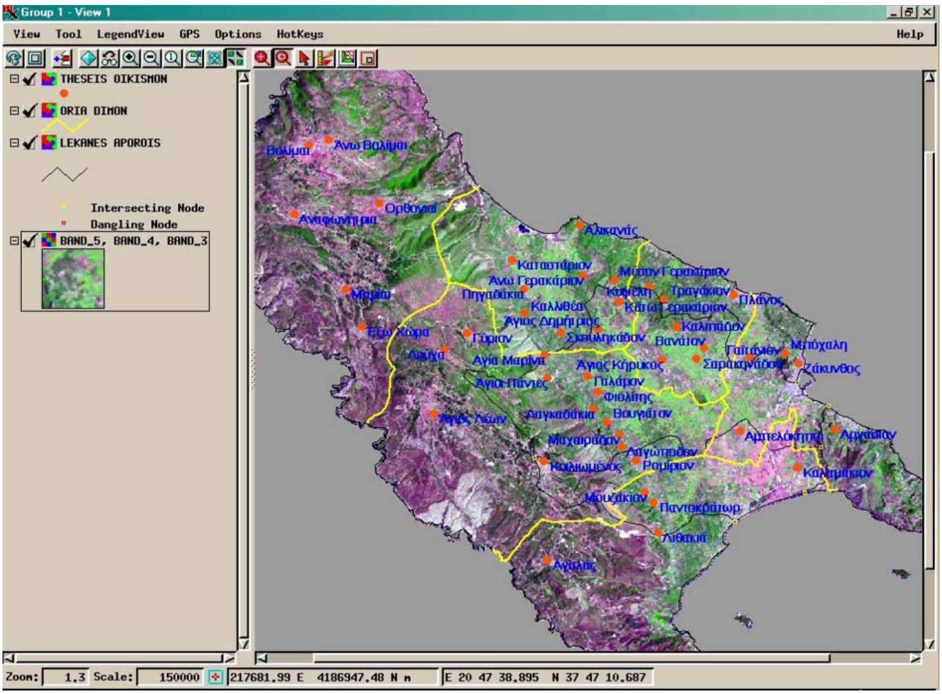


Figure 1: A combination of raster (pseudocolor composite RGB -543 of a Landsat-7 ETM) and vector data (administrative boundaries and water basin boundary) in the GIS environment

2. Adaptation of environmental Information

The main feature of ADAPTIGIS is that it can adapt its interaction with each user. In order to evaluate different information, the system uses a simple decision making model. The suitability of each map for the particular user interacting with the system is estimated taking into account some criteria.

- Degree of interest (i): The values of this criterion show how interesting each information about Zakynthos is for the particular user.
- Need for information (n): This criterion shows how important each information about Zakynthos is for the particular user.

- Comprehensibility of the information(c): This criterion also shows how comprehensible each information about Zakynthos is to the particular user.
- Level of computer skills (l): This criterion shows how comprehensible the way of presentation of each information about Zakynthos is to the particular user.

The values of these criteria are estimated taking into account the information that is stored in the user modeling component of the system. This component stores information about each individual user interacting with the GIS.

For the evaluation of the geographical information, the reasoning mechanism of the system uses the SAW method (Fishburn, 1967, Hwang & Yoon, 1981). According to the SAW method the multi-criteria function is calculated as a linear combination of the values of the four criteria that had identified in the previous experiments:

$U(X_j) = \sum_{i=1}^{4} w_i c_{ij}$, where w_i are the weights of criteria and c_{ij} are the values of the criteria for the X_j geographical information (map).

The criteria used for the evaluation of the geographical information are considered equally important and, therefore, the formula for the calculation of the multi-criteria function is formed:

$$U(X_j) = 0.25i + 0.25n + 0.25c + 0.25l \qquad (1)$$

In view of the values of the multi-criteria function for the different geographical information, the maps are ranked and the one with the highest value is considered to be the most suitable for the user that interacts with the system.

3. Conclusions and future developments

We presented in this paper some implementation issues for the adaptation of geographical information in a GIS, called ADAPTGIS. The ADAPTIGIS is dedicated to the study and management of environmental data. Environmental data are evaluated in terms of some criteria that concern the user needs and skills. More specifically, the system uses a simple decision making model called SAW for ranking different information and select the one that seems most suitable for a user. We plan to extend this work by improving the reasoning mechanism of the system. Furthermore, we intend to developing a GIS for the Web as well as a mobile GIS.

Acknowledgements

The work presented in this paper has been funded by the programme 'ARCHIMEDES II' (Operational Programme for Education and Initial Vocational Training - EPEAEK). The title of the project is 'Adaptive Web Geographical Information System'.

References

[1] Bogdan Moisuc Jérôme Gensel, Paule-Annick Davoine and Hervé Martin: Designing Adaptive Spatio-temporal Information Systems for Natural Hazard Risks with ASTIS 6th International Symposium on

Web & Wireless Geographic Information Systems (W2GIS'2006), December 4-5, Hong Kong, China, 2006.
[2] Choi D-Y (2007) Personalisation local internet in the location-based mobile web search. Decision Support Systems 43, 31-45.
[3] Fishburn, P.C., 1967. Additive Utilities with Incomplete Product Set: Applications to Priorities and Assignments, Operations Research.
[4] Hwang, C.L., Yoon, K., 1981. Multiple Attribute Decision Making: Methods and Applications. Lecture Notes in Economics and Mathematical Systems 186, Springer, Berlin/Heidelberg/New York.
[5] Kabassi, K., Virvou, M., Tsihrintzis G., Kazantzis P.G., Papadatou Th. (2006). Adaptive Presentation of Environmental Information in a GIS. Protection and Restoration of the Environment VIII.
[6] Malpica J.A., Alonso M.C., Sanz M.A. (2007) Dempster-Shafer Theory in geographic information systems: A survey. Expert Systems with Applications, 32, 47-55.
[7] Mathieu Petit, Cyril Ray, Christophe Claramunt: A Contextual Approach for the Development of GIS: Application to Maritime Navigation. W2GIS 2006: 158-169

This page intentionally left blank

Knowledge Acquisition

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-87

Tracing the Parallel Paths of Knowledge Management in the Organizational Domain and the Autonomic Paradigm

Costas Polychronopoulos[a], Makis Stamatelatos[a], Nancy Alonistioti[b]
[a] *Department of Informatics & Telecommunications, University of Athens*
[b] *Department of Informatics, University of Piraeus*

Abstract. Knowledge is a key focus area in the emerging autonomic paradigm. The cognitive control loop inside an autonomic element revolves around the knowledge base built within the element. But the concept sounds familiar. For years now, knowledge epistemology and, in specific, the knowledge management domain have been focusing on the knowledge acquisition, dissemination and sharing mechanisms inside the big picture of knowledge-intensive organizations. This paper argues that the two domains bear significant resemblances and attempts to pinpoint them by forming the necessary analogies.

Keywords. Knowledge management, knowledge dissemination, knowledge information model, autonomic paradigm, intelligent control loop

Introduction

Knowledge is getting increasingly important in all facets of the human activity. Business, economic, social, technical and all other activities are heavily dependent on knowledge. In a globalized environment, knowledge dissemination is of equal importance to the knowledge production itself. Recognizing this fact, the European Union, has started contemplating on the possibility to declare knowledge as the “fifth freedom” in order to facilitate better knowledge diffusion within the European area. The above remarks could not be more true for the emerging autonomic paradigm itself, which dictates that cognitive features are embedded in future computing and network elements. In this environment, knowledge will require more formal and more thorough mechanisms to handle. But the introduction of knowledge management mechanisms in autonomic elements can follow and benefit from the common practices and procedures developed for the organizational knowledge management.

1. Research Directions

1.1. Organizational Knowledge Management

In the highly competitive global operating environment, contemporary organizations have identified knowledge as one of the most important sources of sustainable competitive advantage. The abilities to create, recognize, disseminate and share

knowledge is a definite prerequisite in the continuous organizational pursuit of transforming knowledge into innovation; an effort towards the organization's mission accomplishment [2]. Nevertheless, each one of the above capabilities has proven to require considerable effort in successfully developing them, a fact that has given rise to the emergence of the organizational knowledge management domain.

One of the first tasks the new domain had to deal with was to identify what consisted knowledge within the organizational boundaries. Several definitions and perspectives have been presented in the literature, which approach knowledge from different points of view. In [3] organizational knowledge is defined as "information internalized by means of research, study or experience that has value to the organization". In [4] knowledge is described as "information that is combined with experience, context, interpretation, and reflection. It is a high value form of information that is ready to apply to decisions and actions". Under another point of view which deviates from the system theoretic one, "knowledge resides in the user's subjective context of action based on information [he/she receives]" [5]. A number of other approaches have also focused on subjective factors (e.g. intuitions, hunches, beliefs etc.) in order to explain the knowledge concept sometimes clearly distinguishing it from information [6][10].

Contemplating on the above definitions, a number of interesting issues come up. It is noteworthy that knowledge is frequently related to subsequent decisions and actions. One would argue that knowledge per-se - knowledge for the sake of knowledge - is not in the definition scope of organizational knowledge, where results have to be evident in processes like decision making, action taking etc. Another interesting remark follows the subjective nature suggested in the literature. If it is indeed like that, then *my knowledge is different than yours even if we have been receiving the same information*. One can then easily be led to the conclusion, that the "local" knowledge production is, in general, different than the "global" perception. The process and the intricacies conveying the knowledge produced locally by a knowledge worker to the global organizational environment, while recreating it in new contexts, has been characterized as a "dialectic relationship" [7].

This special attribute of organizational knowledge, namely the fact that it cannot be easily transferred, (e.g. from the "local" to the "global" level), has led to the introduction of the concept of tacit knowledge. Polanyi [1] described it eloquently as the fact that "we can know more that we can tell" giving thus birth to the distinction of knowledge in explicit and tacit. This distinction has been adopted by the work of many knowledge researchers [2][8][9], with some of them even extending it but, still, agreeing on the basic distinction that explicit knowledge is formal and systematic and can be easily communicated and shared whereas tacit knowledge is highly personal, hard to formalize and, therefore, difficult to communicate [2].

Despite the acknowledged difficulties in concretizing and formalizing organizational knowledge, one of the well established means to manage knowledge has been to build knowledge repositories. Building on the proliferation of information technology, repositories consist of business documents, memos, worksheets, databases, workflows etc. and try to capture knowledge that has been developed throughout the organization's past experience, so that it can be shared, reused and applied in new contexts within the organization. Knowledge repositories may exist in more than one levels, reflecting the different levels of knowledge abstraction and detail that can exist in the organizational pyramid [7]. Nevertheless, building the warehouse does not necessarily mean that it is of immediate value for knowledge workers. A number of

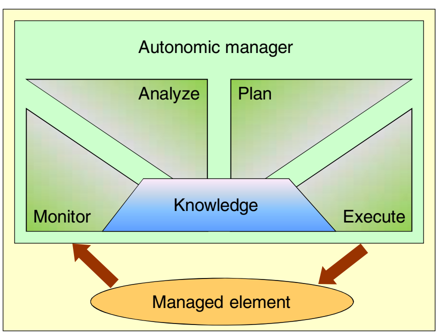
issues, ranging from the repository's search capabilities to the terminological and linguistic differences among its users, can hinder the desirable knowledge sharing. The need for semantically bridging the gap emerges.

1.2. Knowledge Management in the Autonomic Paradigm

What characterizes today's information and communication systems is, mainly, complexity and heterogeneity, at least from a management and control perspective. Indeed, systems become more and more diverse yet interconnected. Such systems have ever increasing requirements in the domain of control, management, configuration and re-configuration, as well, operation and maintenance and optimization.

In a general sense, the essence of autonomic computing systems is self-management, to the capability of computing systems to manage themselves given high-level objectives from administrators. Analyzing that key concept, four aspects of self-management are considered as the basis for autonomic computing [12], namely, self-configuration (ability to automatically setup and (re)adjust the configuration of parameters and resources according to external stimuli), self-optimization (ability to optimize the performance of executed tasks based on experience), self-healing (ability to discover potential malfunctions and recover from failures), and self-protection (ability to ensure overall security and integrity), whereas a number of additional self-* features have been proposed to be supported in the context of an autonomic system, including the concept of self-knowledge/awareness (ability to know the status of its own resources, components, and communications), and self-adaptation (ability to generate and enforce policy rules based on contextual information, thus transforming both itself and the local vicinity) [14] .

An Autonomic system is envisaged to be composed as interactive collections of the so-called *autonomic element* which stands for the basic functional entity of the proposed approach as well. As presented in [11], an autonomic element will integrate one or more Managed Elements as well the Autonomic Manager to realize an automated method for the implementation of the so-called intelligent control loop that is envisaged to be formed through automating a number of key functions [15], as depicted in Figure 1.



Monitor: carries out collection of resource data

Analyze: enables autonomic manager's awareness on the IT environment through correlating and modeling complex situations

Plan: is based on the autonomic element's objectives to form specific actions and tasks towards achieving the mentioned goals

Execute: controls the execution of the plans provided by the Plan function

Knowledge Base: is an implementation of a repository that interacts with the above management functions

Figure 1: The conceptual structure of an autonomic manager [19].

In this context, knowledge in an autonomic system comprises several types of management data such as symptoms, policies, requests for change, and change plans. One step further, the knowledge that is being utilized within an autonomic manager through the monitor-analyze-plan-execute functions shall be a shared knowledge

among different autonomic managers and shall span all four functional features of the autonomic manager.

Knowledge management within autonomic systems is a new and promising topic; the key objective regards knowledge sharing, knowledge re-using and automated knowledge up-dating in a dynamic fashion so as to ensure self-awareness within the autonomic system (i.e. the individual autonomic elements). This can be achieved through integrating new types of knowledge from differentiated resources, formed in different representation schemata in the common knowledge framework. This means that knowledge management in autonomic systems shall address specific issues of knowledge integration, knowledge representation, and knowledge sharing and externalization and this in an increasingly complex environment.

Strassner in [15] identifies a gap in the current approaches in the area of information integration into the managed environment (system/element) in a dynamic fashion. What is proposed is "one way to integrate common management information, regardless of its original source based on building *self-knowledge* into a system and its components, and then using a new ontology-based approach for integrating existing and new knowledge into the management environment, where it can be leveraged using autonomic techniques."

Lewis et al [17] address autonomic network knowledge delivery in a "multicasting" mode with the use of the so-called "*Knowledge Delivery Service* (KDS)", while Sterritt's approach in [18] deals with the awareness of the external environment for effective event management in a complex domain. In such complex environments an event message, which represent a specific type knowledge such as symptoms is not expected to be directly resolved in a specific problem or fault. In this sense, a kind of knowledge integration, specifically the correlation of the event messages, is required for resolving specific symptoms to a certain causal event that may, in turn, trigger autonomic function.

Summarizing the above presented, the concept of knowledge management in autonomic systems can be instantiated in three abstractions:

1. Knowledge is a key issue towards achieving self-management and control in the context of autonomic systems (e.g., decision making, action taking, etc.);
2. Knowledge management within autonomic systems will be realized through knowledge integration, knowledge representation and knowledge sharing;
3. A unified framework is required to address such crucial knowledge management issues in a uniform, extensible way.

2. Identifying the analogies

The autonomic paradigm draws its inspiration from a self-regulating biological system, the human autonomous nervous system. It advocates that the solution for the IT problem of complexity is to introspect and embed autonomic human-like functionalities in the IT systems. Extending this vision and applying it also on the human mental activities (e.g. perception of information, acquisition of knowledge etc.) one can discern some analogies between knowledge developed within an organization environment, handled by humans, and the autonomic paradigm environment, where the autonomic elements handle the acquired knowledge. Highlighting these analogies can help us deepen our understanding for knowledge management in the autonomic systems and progress in a way that will be inspired by the organizational example.

2.1. Recognizing the Importance

Right from the beginning, when the autonomic vision was giving birth to a new technological paradigm, with the publication of IBM's Autonomic Computing Manifesto [19], the introduction of a knowledge base as an integral part of the autonomic element was a proof that knowledge would play a significant role in the workflow, also known as the intelligent control loop, within the autonomic manager. For autonomic elements it is of outmost importance to be able to gather information from the environment they are operating in, thus structuring a picture on the context surrounding them, and use this contextual information to drive the decision-making process in order to adapt to its environment, (re)acting according to its stimuli. This knowledge base is a focal point, where data related to the managed element's environment and the manager's functions (monitor, analyze, plan, execute) is concentrated and stored with the aim to be used within the element itself or disseminated and shared with others. As such, the knowledge base has served in fulfilling the autonomic paradigm requirement for self-awareness.

Similarly, when the term self-knowledge was being articulated in the organizational environment to express the "shared understanding of what the company stands for…" [2], it was already pointing to, what later with the help of information technology acquired a more formal structure; the organization's knowledge repository. It has been repeatedly stressed that organizations would be falling behind the competition, if they could not somehow capture the inherent knowledge. In this sense, self-knowledge provided the means to identify the organization's strengths and weaknesses and, on the other hand, knowledge of the environment served for the organization to adapt to the opportunities and threats posed by its environment. Organizational knowledge management was pointing this way, as the first step that would help with capturing personal knowledge, on the way to succeeding to communicate knowledge to all its human resources, and finally to its products or services. This is the way of an organization to adapt to the requirements of its business environment

2.2. Knowledge Dissemination

The notion of Knowledge Dissemination integrates the knowledge distribution within a reference eco-system in every possible direction thus involving different entities within the same or different level of abstraction in the ecosystem. From a structural point of view the two paradigms that are being considered within this paper, the organizational and the autonomic computing ones, can be considered as "parallel lives". Figure 2, depicts such a parallelism and aims at introducing the key directions of Knowledge Dissemination.

Figure 2 on the left presents the breaking down of tasks and responsibilities within an organization. In the organizational inspiration, an organization, be it a company, can be viewed at the highest abstraction level; on sequence, an organization consists of a number of departments. Being differentiated at the nature of the company, each of the specific departments addresses a (number of) certain aspects within the company framework. Going one level down, each of the afore mentioned departments includes a number of individual workers and the corresponding aspects that have been designated to be addressed by the department are translated into tasks with even more specific scope and objective that are allocated to individual workers.

Proceeding to the autonomic computing parading on the right, Figure 2 extends the architectural approach of [16] that provides a level-based view on the autonomic computing reference architecture. In this, a single autonomic manager is responsible for a specific autonomic capability, one of the previously presented (self-configuring, self-healing etc) whilst in a higher level, the so-called orchestrating autonomic managers are responsible to coordinate autonomic managers working on the same self-* capability or a number of different self-* capabilities.

The following subsections will present the proposed approach on Knowledge dissemination through the behavioral mapping between the two paradigms in the areas of Knowledge Sharing and Transfer.

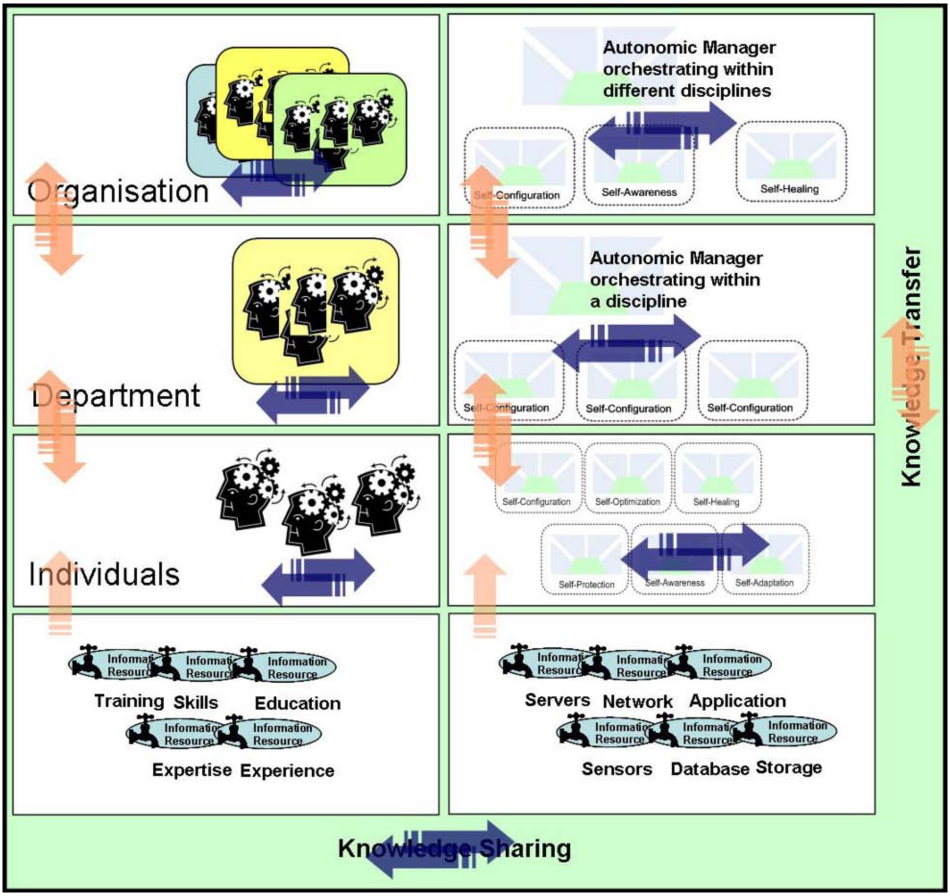


Figure 2: Knowledge management in the organizational environment and the autonomic computing paradigm.

2.2.1. Knowledge Sharing

Following this structured approach on how an autonomic system can be hierarchically composed of individual autonomic managers and elements, one can envisage the knowledge flow among the autonomic managers in each level, referred to herein as Autonomic Knowledge Sharing. Autonomic Knowledge Sharing in the first level of individual autonomic managers consists of the communication among the various

single capabilities autonomic managers in order to share information that might be relevant for potentially more capabilities. An example taken from the network world for a network router is the communication of specific parameters, e.g. the growing number of packets in its buffer, which could in turn trigger different decisions within both the self-configuration and self-optimization autonomic managers. In a second level, knowledge residing in the autonomic managers within one discipline, e.g. self-healing, is shared among them with the help of the orchestrating manager in order to diagnose out of specific symptoms whether a single problem or event has a local or a system-wide scope. Sharing knowledge on this level can help a system discern between an accidental isolated event or a generalized phenomenon that requires system-wide decisions and actions. Finally, on the topmost level, knowledge sharing among managers within different disciplines can facilitate the coordination of event handling across different disciplines. Addressing a system-wide event (e.g. configuration change, fault etc.) might involve more than a single discipline throughout an autonomic system. As an example, sharing knowledge that would require undertaking actions within the self-protection discipline can as well prove beneficial for the self-healing discipline in case the protection mechanisms fail to succeed. It is very likely that the same symptoms that caused the triggering of the self-protection mechanisms have lead the system in a faulty situation, which now it has to recover from with the help of the autonomic self-healing managers.

Similarly, in the organizational environment, knowledge sharing exists in every level of the proposed breaking down. In the first level the individual workers sharing both tacit and explicit knowledge. More specifically, knowledge sharing in this level regards more or less the reaction of each worker to certain triggers that is also visible to other workers or the provided feedback on requests that reflects a worker's background, know-how and way of understanding. Such knowledge is in fact implied among the co-workers in an individual level, not necessarily in the same department. In the second level, knowledge is being shared among the individuals within a certain department towards addressing the department's tasks and achieving the corresponding objectives. Alike, in the third level, the overall organization is presented as an interacting/interworking collection of departments. In this level, knowledge sharing exists among the various departments. It must be also noted that the higher the level of organizational composition is, the more formal is the way of knowledge sharing.

2.2.2. Knowledge Transfer

In the second direction of the Knowledge Dissemination, the vertical one, knowledge transfer exists between two consecutive levels and, in a more abstract way, among every level. Argote & Ingram define knowledge transfer as "the process through which one unit is affected by the experience of another" [20]. Given the structural approach of Figure 2, Knowledge Transfer as realized among the various levels of abstraction that have been identified can be considered as a means of gradually building the self-knowledge within the reference system.

In an organizational environment individual workers feed with knowledge their departments and each department feeds the organization as a whole in an attempt to gradually form a comprehensive picture of the organization's modus operandi as a whole. In the reverse direction, knowledge acquired and produced in the upper level of administration is also pushed downwards. In the downward direction though, knowledge is more about the operating environment and its communication towards the

lower levels deals with the translation internally in terms of the effects this has on the organization's internal processes. This knowledge transfer is being carried out in an official and well-defined way that is interpretable by every of the two physical or logical parts of the transfer, be it meetings, reports, demonstrations, analyses, etc.

In the autonomic computing paradigm it can be considered that every lower level provides knowledge to its higher level of abstraction by using a common reference knowledge representation scheme in order for the knowledge to be understandable throughout the autonomic system breaking down. This implies the requirement for a common approach of knowledge modeling that will be developed in a representation – agnostic way. Such a common approach is initiated in the next section. Additionally, as mentioned knowledge transfer being considered as a step prior to self-awareness building can be modeled as presented in Figure 3. In this, every level of the breaking down approach (as presented in Figure 2) contributes to self awareness that resides in the top of the layered approach. In Figure 3 the self knowledge building is presented through a "building up" approach. This approach depicts each lower level contribution to the higher (physical) and to the overall self-knowledge (logical). More specifically, Level 0 (Information Sources) feeds Level 1 with structured information from the source-come raw data. In turn, each of the individual autonomic managers provides the Level 2 (first level of orchestration – view point specific) with knowledge leveraged from each individual autonomic manager's viewpoint; in this sense Level 2 is fed with multiple viewpoint-specific knowledge "packages". Alike, each orchestrating autonomic manager (within a certain discipline) feeds Level 3 with discipline-specific knowledge and, finally, Level 3 ties up the self-knowledge building within an autonomic element by integrating several discipline-specific knowledge packages.

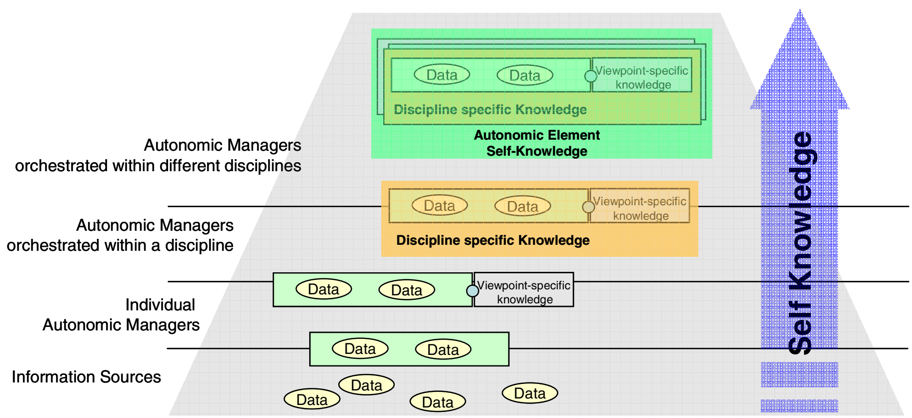


Figure 3: Knowledge Transfer - Self-Knowledge building.

From a perspective of the proposed approach, if such coordination is translated in the domain of knowledge, it regards the knowledge sharing among peer level autonomic managers in the horizontal direction, as well as the knowledge transfer between different levels of the reference architecture in the vertical direction.

2.3. Towards a Common Information Model

Information Modeling is a specific area of the data modeling that addresses aspects related to the analysis, the design, and the representation of the information within a

system. In this sense, information models provide formal representation of the information to be managed and exchanged within the scope of a system. Key issue in information modeling is to provide formal representation without implying a mapping to a specific implementation approach but supporting the extraction of different mapping to specific implementation solutions be it software platforms and/or software vendors.

Expanding the scope of information modeling, and based on the previous section analysis, a unified approach seems to be required for knowledge sharing and representation in autonomic systems. As argued in [15] "there will most likely never be a single unified information model but there must be an extensible modeling basis for integrating diverse data in a knowledge management framework in autonomic systems".

In the organizational paradigm, such a unified approach exists within an organization in terms of a common language for expressing and communicating the business goals and achievements, the management information and constraints. Alike, based on the hierarchical approach (Figure 2) a business objective or management information is communicated to the various domains and then down to the individual workers. Depending on each individual worker's skills and capabilities such information can be:

- Directly or indirectly (through "on the fly" translation) interpreted, in the sense of grammar and syntax, depending on whether the business language is well known to the worker (native or advanced speaker)
- Directly or indirectly interpreted, depending on the technical expertise of the worker: for instance, a worker may carry out a task either individually, or being supported by a more experienced colleague of his/her.

Coming back to autonomic computing paradigm, several information types are needed to be taken into account in order for an autonomic "thing", be it an autonomic manager, an autonomic element or a system to be self-managed. More specifically, a resource of information stands for the ever-changing environment of an autonomic thing; additionally, information about the goals/objectives that need to be achieved and the constraints to which it needs to be aligned are also included in the autonomic computing information space. In this sense, several stakeholders' policies must be also interpreted thus need to be represented in a way that enables co-interpretation with contextual information. A system element may be required to adapt to environmental changes but also to business requirements.

3. Conclusions and Future Research

This paper has attempted to bring forward the analogies existing in the organizational environment and the autonomic paradigm as far as the knowledge domain is concerned. Starting with the consensus built around the importance of knowledge management in both domains, a thorough analysis on the knowledge dissemination facets and intricacies has been presented to conclude with the need for a common knowledge information model.

An interesting point for further research would be to draw from this similarity remark and be inspired to transfer ideas and practices developed within the organizational environment to the newly born autonomic paradigm. The authors have identified a potential application in the introduction of new autonomic elements into operation and the way they can benefit from neighboring elements' acquired

knowledge following the common practice of the apprenticeship applied in the organizational domain.

Acknowledgement

This work was performed in project E3 which has received research funding from the Community's Seventh Framework programme. This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein. The contributions of colleagues from E3 consortium are hereby acknowledged.

References

[1] Polanyi, M., "The Tacit Dimension", Smith (1983), ISBN: 0844659991
[2] Nonaka, I., "The Knowledge-Creating Company", Harvard Business Review (HBR Classic), July 2007.
[3] Kerssens-Van Drongelen, I.C., de Weerde-Nederhof, P.C. and Fissher, O.A.M. (1996), ''Describing the issues of knowledge management in R&D: toward a communications and analysis tool'', R&D Management, vol. 26 no. 3, pp. 213-30. doi:10.1111/j.1467-9310.1996.tb00957.x
[4] Davenport, T.H., de Long, D.H. and Beers, M.C. (1998), ''Successful knowledge management projects'', Sloan Management Review, Vol. 39 No. 2, pp. 43-57.
[5] Malhotra, Y., Deciphering the Knowledge Management Hype, Journal for Quality & Participation, July-August, 1998.
[6] Dretske, F. (1981), Knowledge and the Flow of Information, MIT Press, Cambridge, MA.
[7] Davis, Joseph G.; Subrahmanian, Eswaran; Westerberg, Arthur W., "The "global" and the "local" in knowledge management", Journal of Knowledge Management, 2001, vol.9 No.1, pp. 101-112, http://www.emeraldinsight.com/10.1108/13673270510582992
[8] Stenmark, D., "Turning tacit knowledge tangible", System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on , vol., no., pp. 9 pp. vol.2-, 4-7 Jan. 2000.
[9] Leonard, D. and Sensiper, S., "The Role of Tacit Knowledge in Group Innovation", California Management Review, vol. 40, no. 3, 1998, pp. 112-132.
[10] McDermott R., "Why Information Technology Inspired but Cannot Deliver Knowledge Management", California Management Review, vol. 41 no. 4, 1999, pp. 103-117.
[11] IBM, "Autonomic Computing: IBM's perspective on the State of Information Technology", available at: http://www-1.ibm.com/industries/goverment/doc/content/resource/though/278606109.html
[12] Kephart, O., Jeffrey, Chess, M., David, "The Vision of Autonomic Computing", IEEE Computer, vol. 36, no.1, 2003, pp. 41-50. doi: 10.1109/MC.2003.1160055
[13] Stamatelatos M., "The E2R II Business Outlook: Framework, Instantiations and Challenges for Reconfigurability", E2R II White Paper, October 2006, http://e2r2.motlabs.com/whitepapers/E2RII_UBM_White_Paper.pdf
[14] Z. Boufidis, N. Alonistioti, M. Stamatelatos, D. Bourse, K. El-Khazen, M. Muck, "Evolution of the Reconfiguration Management Plane for Autonomic Communications", IST Summit 06, Mykonos, Greece
[15] Strassner, J., "Knowledge Management Issues for Autonomic Systems"
[16] IBM, "An architectural blueprint for autonomic computing", IBM Autonomic Computing White Paper, 3rd ed., June 2005.
[17] Lewis, D., O'Sullivan, D., Power, R., and Keeney, J., "Semantic Interoperability for an Autonomic Knowledge Delivery Service", Proceedings of the 2nd IFIP TC6 International Workshop on Autonomic Communication (WAC 2005), Vouliagmeni, Athens, Greece, 2005.
[18] Sterrit, R., "Towards Autonomic Computing: Effective Event Management", Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop, IEEE 2003.
[19] Horn, P., "IBM Autonomic Computing Manifesto", October 2001, http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf
[20] Argote, L., P. Ingram (2000). "Knowledge transfer A Basis for Competitive Advantage in Firms." Organizational Behavior and Human Decision Processes 82(1): 150-169.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-97

Deontic Relevant Logic in Knowledge-based Requirements Engineering

Jingde CHENG [1]
Department of Information and Computer Sciences, Saitama University, Japan

Abstract. Requirements engineering is inherently concerned with discovering and/or predicting purposes, goals, and objectives of software systems. To discover/predict, analyze, elicit, specify, and reason about various requirements of software systems, we need a right fundamental logic system to provide us with a logical validity criterion of reasoning as well as a formal representation and specification language. This short position paper briefly shows that deontic relevant logic is a hopeful candidate for the fundamental logic we need.

Keywords. Discovery, Prediction, Relevant Reasoning, Normative reasoning, Strong relevant logic

1. Introduction

Requirements engineering is inherently concerned with discovering and/or predicting purposes, goals, and objectives of target software systems [5, 12]. Some inherent difficulties in requirements engineering are: (1) customers/end-users of target systems do not necessarily understand and describe (even in natural languages) their desires explicitly, (2) sometimes there are some conflict among customers/end-users of target systems, (3) customers/end-users of target systems may modify even change their desires time after time during requirement analysis and definition, (4) the data/information about the environment in which the target systems will work may be incomplete and even inconsistent, (5) non-functional requirements may be more critical than functional requirements but they are usually difficult to realize and often recognized at later stage of requirement analysis and definition, and (6) as the final results of requirements engineering, the requirements documents of target systems must be not only precise, detailed, technical, at the same time, but also readable, understandable, to domain experts and customers/end-users of target systems who are not so familiar software engineering, and so on. Knowledge-based requirements engineering intends to deal with these difficulties by invoking the power of knowledge representation and reasoning [4, 8]. However, it seems that those approaches proposed in various researches of knowledge-based requirements engineering did not considerably reduce inherent difficulties in requirements engineering. Today, requirements engineering is still the most ad hoc and informal area in software engineering in the sense that formal, systematic, and theoretical methodologies for activities and tasks in requirements engineering have not been established well.

[1] Corresponding Author: Department of Information and Computer Sciences, Saitama University, Saitama, 338-8570, Japan; E-mail: cheng@ics.saitama-u.ac.jp

To discover/predict, analyze, elicit, specify, and reason about various requirements of software systems, we need a right fundamental logic system to provide us with a logical validity criterion of reasoning as well as a formal representation and specification language. The question, "Which is the right logic?" invites the immediate counter-question "Right for what?" Only if we certainly know what we need, we can make a good choice. Thus, what are essential requirements for the fundamental logic to underlie knowledge-based requirements engineering? This short position paper proposes some essential requirements for the fundamental logic to underlie knowledge-based requirements engineering, briefly discusses why classical mathematical logic, its various classical conservatives extensions, or its non-classical alternatives are not suitable candidates for the fundamental logic, and briefly shows that deontic relevant logic is a more hopeful candidate for the fundamental logic we need.

2. Basic Notions

Reasoning is the ***process*** of drawing ***new conclusions*** from given premises, which are already known facts or previously assumed hypotheses to provide some ***evidence*** for the conclusions. Therefore, reasoning is intrinsically ampliative, i.e., it has the function of enlarging or extending some things, or adding to what is already known or assumed. In general, a reasoning consists of a number of arguments ***in some order***. An ***argument*** is a set of ***statements*** (or ***declarative sentences***) of which one statement is intended as the ***conclusion***, and one or more statements, called "***premises***," are intended to provide some evidence for the conclusion. An argument is a conclusion standing in relation to its supporting evidence. In an argument, a claim is being made that there is some sort of ***evidential relation*** between its premises and its conclusion: the conclusion is supposed to ***follow from*** the premises, or equivalently, the premises are supposed to ***entail*** the conclusion. Therefore, the correctness of an argument is a matter of the ***connection*** between its premises and its conclusion, and concerns the ***strength*** of the relation between them. A ***logically valid reasoning*** is a reasoning such that its arguments are justified based on some ***logical validity criterion*** provided by a logic system in order to obtain correct conclusions. Today, there are so many different logic systems motivated by various philosophical considerations. As a result, a reasoning may be valid on one logical validity criterion but invalid on another. For example, the ***classical account of validity***, which is one of fundamental principles and assumptions underlying classical mathematical logic and its various conservative extensions, is defined in terms of ***truth-preservation*** (in some certain sense of truth) as: an argument is valid if and only if it is impossible for all its premises to be true while its conclusion is false. Therefore, a classically valid reasoning must be ***truth-preserving***. On the other hand, for any correct argument in scientific reasoning as well as our everyday reasoning, its premises must somehow be ***relevant*** to its conclusion, and vice versa. The ***relevant account of validity*** is defined in terms of ***relevance*** as: for an argument to be valid there must be some connection of meaning, i.e., some relevance, between its premises and its conclusion. Obviously, the relevance between the premises and conclusion of an argument is not accounted for by the classical logical validity criterion, and therefore, a classically valid reasoning is not necessarily relevant.

Proving is the process of finding a justification for an explicitly ***specified statement*** from given ***premises***, which are already known facts or previously assumed hypotheses to provide some ***evidence*** for the specified statement. A ***proof*** is a

description of a found justification. A ***logically valid proving*** is a proving such that it is justified based on some logical validity criterion provided by a logic system in order to obtain a correct proof. The most intrinsic difference between reasoning and proving is that the former is intrinsically prescriptive and predictive while the latter is intrinsically descriptive and non-predictive. The purpose of reasoning is to find some new conclusion previously unknown or unrecognized, while the purpose of proving is to find a justification for some specified statement previously given. Proving has an explicitly given target as its goal while reasoning does not.

Discovery is the process to find out or bring to light of that which was previously unknown. For any discovery, the discovered thing and its truth must be unknown before the completion of discovery process. ***Prediction*** is the action to make some future event known in advance, especially on the basis of special knowledge. It is a notion must relate to a point of time to be considered as the reference time. For any prediction, both the predicted thing and its truth must be unknown before the completion of that prediction. Because reasoning is the only way to draw new, previously unknown conclusions from given premises, there is no discovery/prediction process that does not invoke reasoning. Moreover, because no discovery/prediction has a previously explicitly given target as its goal, any meaningful reasoning in any discovery/prediction process should be relevant, i.e., its premises must somehow be relevant to its conclusion, and vice versa.

In logic, a sentence in the form of 'if ... then ...' is usually called a ***conditional proposition*** or simply ***conditional*** which states that there exists a relation of sufficient condition between the 'if' part and the 'then' part of the sentence. In general, a conditional must concern two parts which are connected by the connective 'if ... then ...' and called the ***antecedent*** and the ***consequent*** of that conditional, respectively. The truth of a conditional depends not only on the truth of its antecedent and consequent but also, and more essentially, on a necessarily relevant and conditional relation between them. From the viewpoint of object logic, there are two classes of conditionals. One class is empirical conditionals and the other class is logical conditionals. For a logic, a conditional is called an ***empirical conditional*** of the logic if its truth-value, in the sense of that logic, depends on the contents of its antecedent and consequent and therefore cannot be determined only by its abstract form; a conditional is called a ***logical conditional*** of the logic if its truth-value, in the sense of that logic, depends only on its abstract form but not on the contents of its antecedent and consequent, and therefore, it is considered to be universally true or false. A logical conditional that is considered to be universally true, in the sense of that logic, is also called an ***entailment*** of that logic. Indeed, the most intrinsic difference between various different logic systems is to regard what class of conditionals as entailments.

3. The Logic Basis of Knowledge-based Requirements Engineering

First of all, for any activity/task in requirements engineering, there is no explicitly given target as its goal; it is rather the goal that is the target the activity/task should find. Therefore, what is the way requirements engineering has to invoke is reasoning rather than proving. The present author claims that one should consider the following essential requirements for the fundamental logic to underlie knowledge-based requirements engineering. First, as a general logical criterion for the validity of reasoning, the logic must be able to underlie relevant reasoning as well as truth-

preserving reasoning in the sense of conditional, i.e., for any reasoning based on the logic to be valid, if its premises are true in the sense of conditional, then its conclusion must be relevant to the premises and true in the sense of conditional. Second, the logic must be able to underlie ampliative reasoning in the sense that the truth of conclusion of the reasoning should be recognized after the completion of the reasoning process but not be invoked in deciding the truth of premises of the reasoning. From the viewpoint to regard reasoning as the process of drawing new conclusions from given premises, any meaningful reasoning must be ampliative but not circular and/or tautological. Third, the logic must be able to underlie paracomplete reasoning and paraconsistent reasoning. In particular, the so-called principle of Explosion that everything follows from a contradiction cannot be accepted by the logic as a valid principle. In general, our knowledge about a domain may be incomplete and/or inconsistent in many ways, i.e., it gives us no evidence for deciding the truth of either a proposition or its negation, and/or it directly or indirectly includes some contradictions. Therefore, reasoning with incomplete and/or inconsistent knowledge is the rule rather than the exception in our everyday lives and almost all scientific disciplines. Finally, because in requirements engineering we often need to distinguish between what ought to be done and what is really the case, a formalisation of normative notions is necessary, i.e., logic must be able to underlie normative reasoning.

Almost all traditional formal methods in software engineering are somehow based on classical mathematical logic (**CML** for short) or its various classical conservatives extensions. This approach, however, may be suitable to searching and describing a formal proof of a previously specified statement or rule, but not necessarily suitable to reasoning a new fact, inventing a new concept, and discovering a new statement or rule because the aim, nature, and role of **CML** is descriptive and non-predictive rather than prescriptive and predictive. For requirements engineering, any formal method based on **CML** is inherently not suitable to activities/tasks that concerned with reasoning rather than proving. **CML** was established based on a number of fundamental assumptions. Among them, the most characteristic one is the classical account of validity that is the logical validity criterion of **CML** by which one can decide whether the conclusion of an argument really does follow from its premises or not in the framework of **CML**. However, because the relevance between the premises and conclusion of an argument is not accounted for by the classical validity criterion, a reasoning based on **CML** is not necessarily relevant. On the other hand, in **CML** the notion of conditional, which is intrinsically intensional but not truth-functional, is represented by the notion of material implication, which is intrinsically an extensional truth-function. This leads to the problem of 'implicational paradoxes' [1, 2] as well as the problem that a reasoning based on **CML** must be circular and/or tautological but not ampliative. Moreover, because **CML** accepts the principle of Explosion, reasoning under inconsistency is impossible within the framework of **CML**. Finally, as a tool to describe ideal mathematical proofs, **CML** does not distinguish between ideal states and actual states and cannot to underlie normative reasoning. Therefore, **CML** cannot satisfy any of the essential requirements for the fundamental logic to underlie knowledge-based requirements engineering. The above facts are also true to those classical conservative extensions or non-classical alternatives of **CML** where the classical account of validity is adopted as the logical validity criterion and the notion of conditional is directly or indirectly represented by the material implication.

Deontic logic is a branch of philosophical logic to deal with normative notions such as obligation (ought), permission (permitted), and prohibition (may not) for

underlying normative reasoning [3]. Informally, it can also be considered as a logic to reason about ideal versus actual states or behaviour. It seems to be an adequate tool to represent and reason about requirements. However, because any classical deontic logic is a classical conservatives extension of **CML**, all problems in **CML** caused by the classical account of validity and the material implication also remained in classical deontic logic. Moreover, there is the problem of deontic paradoxes in classical deontic logic [3, 13].

Traditional relevant (or relevance) logics ware constructed during the 1950s in order to find a mathematically satisfactory way of grasping the elusive notion of relevance of antecedent to consequent ins conditionals, and to obtain a notion of implication which is free from the so-called 'paradoxes' of material and strict implication [1, 2]. A major characteristic of the relevant logics is that they have a primitive intensional connective to represent the notion of (relevant) conditional and their logical theorems include no implicational paradoxes. The underlying principle of the relevant logics is the relevance principle, i.e., for any entailment provable in a relevant logic, its antecedent and consequent must share a propositional variable. Variable-sharing is a formal notion designed to reflect the idea that there be a meaning-connection between the antecedent and consequent of an entailment. It is this relevance principle that excludes those implicational paradoxes from logical axioms or theorems of relevant logics. Also, since the notion of entailment is represented in the relevant logics by a primitive intensional connective but not an extensional truth-function, a reasoning based on the relevant logics is ampliative but not circular and/or tautological. Moreover, because the relevant logics reject the principle of Explosion, they can certainly underlie paraconsistent reasoning. However, although the traditional relevant logics have rejected those implicational paradoxes, there still exist some logical axioms or theorems in the logics, which are not natural in the sense of conditional. Therefore, in the framework of any traditional relevant logics, even if a reasoning is relevantly valid, neither the truth of its conclusion in the sense of conditional nor the relevance between its premises and conclusion can be guaranteed necessarily. This situation is the same as that in **CML**.

In order to establish a satisfactory logic calculus of conditional to underlie relevant reasoning, Cheng has proposed some strong relevant (or relevance) logics [6, 7]. The logics require that the premises of an argument represented by a conditional include no unnecessary and needless conjuncts and the conclusion of that argument includes no unnecessary and needless disjuncts. What underlies the strong relevant logics is the strong relevance principle: If A is a theorem of a strong relevant logic, then every propositional variable in A occurs at least once as an antecedent part and at least once as a consequent part. Since the strong relevant logics are free of not only implicational paradoxes but also conjunction-implicational and disjunction-implicational paradoxes, in the framework of strong relevant logics, if a reasoning is valid, then both the relevance between its premises and its conclusion and the validity of its conclusion in the sense of conditional can be guaranteed in a certain sense of strong relevance. However, strong relevant logics cannot underlie normative reasoning.

Consequently, what we need is a suitable combination of strong relevant logics and classical deontic logic such that it can satisfy all the essential requirements for the fundamental logic system to underlie knowledge-based requirements engineering. The deontic relevant logics [7, 13] can satisfy all the essential requirements for the fundamental logic system to underlie knowledge-based requirements engineering. First, the deontic relevant logics provide a formal language with normative notions which

can be used as a formal representation and specification language for representing and specifying requirements. Second, the deontic relevant logics provide a sound logical basis for reasoning about requirements. Based on the logics, truth-preserving and relevant reasoning in the sense of conditional, ampliative reasoning, paracomplete reasoning, paraconsistent reasoning, and normative reasoning are all possible. There is no other logic system proposed for knowledge-based requirements engineering has these advantages.

4. Concluding Remarks

Classical deontic logic has been used in Computer Science, in particular, in requirements engineering from 1993 [9-11]. The propositional deontic relevant logic was first proposed by Tagawa and Cheng to solve the well-known problem of deontic paradoxes in classical deontic logic [13], and the idea to use deontic relevant logic in Computer Science was first proposed by Cheng in 2003. The work presented in this short position paper is our first step for establishing a fundamental logic basis to underlie knowledge-based requirements engineering. There are some challenging theoretical and technical problems that have to be solved in order to apply the deontic relevant logics to practices in the real world.

References

[1] A. R. Anderson, and N. D. Belnap Jr., *Entailment: The Logic of Relevance and Necessity, Vol. I*, Princeton University Press, Princeton, 1975.

[2] A. R. Anderson, N. D. Belnap Jr., and J. M. Dunn, Entailment: *The Logic of Relevance and Necessity, Vol. II*, Princeton University Press, Princeton, 1992.

[3] L. Aqvist, "Deontic Logic," In: D. Gabbay and F. Guenthner (Eds.), *Handbook of Philosophical Logic, 2nd Edition, Vol. 8*, pp. 147-264, Kluwer Academic, 2002.

[4] A. Borgida, S. Greenspan, and J. Mylopoulos, "Knowledge Representation as the Basis for Requirements Specifications," Computer, Vol. 18, No. 4, pp. 82-91, IEEE-CS Press, 1985.

[5] B. H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering," *Proc. Conference on the Future of Software Engineering*, pp. 285-303, IEEE-CS Press, 2007.

[6] J. Cheng, "A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery," In: E. Kawaguchi, H. Kangassalo, H. Jaakkola, and I. A. Hamid (Eds.), *Information Modelling and Knowledge Bases XI,* pp. 136-159, IOS Press, Amsterdam, 2000.

[7] J. Cheng, "Strong Relevant Logic as the Universal Basis of Various Applied Logics for Knowledge Representation and Reasoning," In: Y. Kiyoki, J. Henno, H. Jaakkola, and H. Kangassalo (Eds.), *Information Modelling and Knowledge Bases XVII,* pp. 310-320, IOS Press, Amsterdam, 2006.

[8] A. J. Czuchry, Jr. and D. R. Harris, "KBRA: A New Paradigm for Requirements Engineering," Intelligent Systems, Vol. 3, No. 4, pp. 21-35, IEEE-CS Press, 1988.

[9] E Dubois, "Use of Deontic Logic in the Requirements Engineering of Composite Systems," In: J.-J. C. Meyer, and R. Wieringa (Eds.), "Deontic Logic in Computer Science: Normative System Specification," pp. 125-139, John Wiley & Sons, 1993.

[10] J.-J. C. Meyer, and R. Wieringa (Eds.), "Deontic Logic in Computer Science: Normative System Specification," John Wiley & Sons, 1993.

[11] J.-J. C. Meyer, R. Wieringa, and F. P. M Dignum, "The Role of Deontic Logic in the Specification of Information Systems," in J. Chomicki and G. Saake (Eds.), "Logics for Databases and Information Systems," pp. 71-115. Kluwer Academic, 1998.

[12] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," Proc. Conference on the Future of Software Engineering, pp. 35-46, ACM Press, 2000.

[13] T. Tagawa and J. Cheng, "Deontic Relevant Logic: A Strong Relevant Logic Approach to Removing Paradoxes from Deontic Logic," Lecture Notes in Artificial Intelligence, Vol. 2417, pp. 39-48, Springer-Verlag, Berlin, 2002.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-103

The Lecture Contents with the Captions by Using Meaningful Reception Learning

Rihito YAEGASHI [a,1], Kazuhiro UENOSONO [b] Hedezi ENOKIZU [b] and Seiichi KOMIYA [b]

[a] *Shibaura Instuiteu of Technology, Center for Information Science*
[b] *Shibaura Institute of Technology*

Abstract. There are some programs which give the Malaysian students the pre-education for studying in Japan such as JAD program （Japan Associate Degree Program） in Malaysia. In these programs, it is impossible for the lecturers who are dispatched from Japan to cover all the majors. Therefore, they often give some lessons by the distance learning from Japan, but they have not got the sufficient educational effect yet. Distance learning are the real-time lecture which Japanese lectures give to Japanese students, or the lecture contents which just record these lectures. It is very difficult for the students at JAD programs who are not Japanese natives to understand all the contents of these distance learning. From these background, JAD Program makes the lecture contents with the captions for all the sentences, and conducts the distance learning.

This paper discusses the method to relate with the background information by using the Advanced organizer which D.P.Ausubel proposed. Also, this paper describes the interface technology which uses the visual effect of the captions to express the relationship of Advanced organizer. The authors are conducting the distance learning, using these lecture contents which are based on these methods. The authors aim to inspect the effect of these lecture contents.

Keywords. Distance Learning, Caption contents, Skimming,

Introduction

JAD Program, Japan Associate Degree Program[1] is the pre-education program for Malaysian oversea students. It was set up based on Look East Policy which was advocated by former Malaysian prime minister, DR.MAHATHIR. The feature of JAD Program is to teach engineering to the students. They study not only Japanese but also basic science subjects such as Physics, Mathematics, Chemistry and engineering subjects. These subjects which are equivalent for the education at the university are taught in Japanese. In the pre-education programs including JAD Program, it is impossible to cover all the special areas only by the teachers dispatched from Japan.. Therefore, some lectures are often shown as the distance lecture from Japan. The students in JAD Program need to take the distance lecture in Japanese after they have studied Japanese for one year.

[1] Shibaura Institute of Technology, Center for Information Science, Toyosu 3-7-5, Koutou-ku Tokyo 135-8548, Japan; E-mail: rihito@shibaura-it.ac.jp.

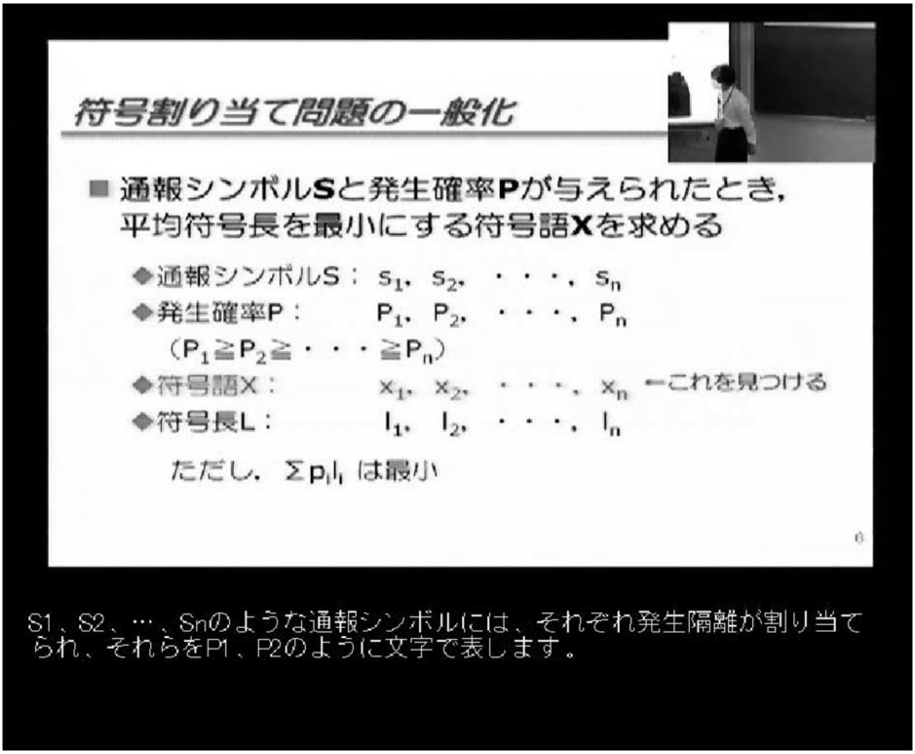


Figure 1. The Lecture Contents with the Captions.

Distance learning are the real-time lectures which Japanese teachers give to Japanese students, or the lecture contents which just record these lectures. It is very difficult for the students who have studied Japanese only for one year to understand all the contents of these distance learning. Language is one of the issues when we send the lecture contents to abroad. The teachers who usually give the lectures to the students in Malaysia can use the wards which the students can understand, because they can grasp how far the students understand Japanese from their situation. However, when we use the distance learning, even if it is the real-time lecture, it is actually impossible for the teacher to grasp the students'situation in Malaysia. In JAD Program, comparing with the general lessons, we can not get enough educational effect with the distance learning, because of the lack of the students'ability for Japanese and the characteristics of the distance learning.

Considering these backgrounds, we need to compensate for the students'ability for Japanese. JAD Program has conducted the distance learning which the teachers record for JAD Program, and add the captions for all the teacher's utterance. Figure 1 is the lecture contents with the captions which are made for JAD Program. The lecture contents are made as RealMedia, and the captions are made as RealText. The distance learning is played back with the lecture contents, the captions and SMIL at the same time[2,3]. Generally speaking, people understand that the people's reading style is changing recently. The previous style was that we read something from the first page to the last page[4]. However, we read only the important points now. This style is called "Skimming". Some

people experimented on it, and they clarified that people are not good at the Skimming from the display, comparing with the paper materials[5]. In this situation, the skimming means the action that the people understand the meaning of the text in a short time. The authors aimed to make the captions that the students can skim correctly in a short time. In order to advance the students'understanding by using the lecture contents with the skimming, this paper discusses the method to relate with the background information by using the Advanced organizer.[6] which D. P. Ausubel proposed. Also, this paper describes the interface technology which uses the visual effect of the captions to express the relationship of Advanced organizer.

1. The relationship with the background information by using the Expository Organizer

D. P. Ausubel contradicted the opinion that the lectures make the students negative and passive. He insisted, "the process which the people understand the knowledge and accept it is the positive process. Therefore, it is possible to make the strong linkage between the professors and the learning. " From this opinion, he suggested Meaningful Reception Learning[6]. D. P. Ausubel proposed the Advance Organizer as the concrete act from the teachers on Meaningful Reception Learning. The Advance Organizer is the method that the teacher tells a short introduction before the students start reading the text. The Advance Organizer has two kinds, the Expository Organizer and the comparative organizer. The teachers use either the Expository Organizer and the comparative organizer, according to the level of the students about the learning contents. When the learning contents are very new to the students and they are not familiar to the learning contents, the expository organizer advances to relate the students'background information and the learning contents, commenting on the idea which is related to the learning contents. The teachers use the The comparative organizer, when the students already have some background information about the learning contents. Compared the concept which the students already have with the concept which they will learn in the future, the Comparative Organizer clarifies the similarity and the difference between them, and prevents the students'misunderstanding. Generally speaking, the university education is regarded as the process to repeat gathering new knowledge based on the background knowledge. Then, the authors planed to use the comparative organizer for the knowledge which the students have already learned, and use the Expository Organizer for the knowledge which the students have not learned yet.

2. The interface technology using the visual effect

Generally speaking, people understand that the people's reading style is changing recently[4]. The previous style was that we read something from the first page to the last page. However, we read only the important points now. This style is called "skimming". Some people experimented on it, and they clarified that people are not good at the skimming from the display, comparing with the paper materials [5]. In this paper, the skimming means the action that the people understand the meaning of the text in a short time. The interface technology which uses the visual effect and the text summarizing

technology are known as the studies on supporting reading from the display. The interface technology which uses the visual effect has the Overview+detail effect, the Fisheye effect and the Linear effect. The Overview+detail is showing display of the entire list. It enables the students to understand which part of the display they are looking at. The Fisheye makes the notable parts (focus) big, and makes other parts (context) small. The students can immediately understand the notable parts. The Liner effect is the method which the teacher arranges the sentences in order to the wards. [8]. The Overview+detail effect supports the students'understanding, but it takes more time than other methods. The Fisheye effect is known as the method which is effect for critical tasks[7]. The text summarizing technology assists the students'reading to reduce the amount of the documents. However, some people point out the problem of the consistency when the text summarizing technology automatically makes the summary.[9,10]. Most of the studies on summarizing the dialogue-style text make the approach based on the summary which is presented by the method of distilling the important sentences. There is a problem that it takes time to make the summary, because the summary made in these researches is a summary based on the reader's preference. Then, the authors express the relationship of the Advance Organizer as the captions, using the Fisheye effect and the Liner effect from the interface technology on this study.

3. The method of making the lecture contents with the captions by using the Meaningful Reception Learning

The teacher needs to submit the summary and the key wards in advance when he wants to make the lecture contents with the captions. He needs to submit the summary and the key wards for each of the comparative organizer and the Advance Organizer. He itemizes the summary. After the lecture, he divides the utterances into the comparative organizer and the Advance Organizer. The comparative organizer and the advance organizer have different colors and font sizes on the captions. The comparative organizer is shown by blue, and the Advance Organizer is shown by red. The text sizes are also big in this lecture contents with the captions which uses Meaningful Reception Learning.

4. Conclusion

In order to advance the students'understanding by using the lecture contents with the skimming, this paper discussed the method to relate with the background information by using the Advanced organizer which D.P.Ausubel proposed. Also, this paper described the interface technology which uses the visual effect of the captions to express the relationship of Advanced organizer. The authors are conducting the distance learning, using these lecture contents which are based on these methods. The authors aim to inspect the effect of these lecture contents.

Acknowledgement

This study is supported by The Plan of Virtual-One Campus by Shibaura Institute of Technology. It is one of the cyber campus schemes which is advocated by the Minister of Education in 2006. We appreciate their cooperation.

References

[1] JUCTe Higher Education Loan Project:HELP(HELP)・HELP3 https://office.shibaura-it.ac.jp/kokusai/jucte/program/help3.html

[2] RealNetworks, Inc, http://www.realnetworks.com/

[3] Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, W3C Recommendation, 15-June-1998, http://www.w3.org/TR/REC-smil/

[4] D.M. Levy;I read the news today, oh boy: reading and attention in digital libraries, Proceedings of the second ACM international conference on Digital libraries, p202-211, 1997

[5] P.Muter and P.Maurutto: Reading and skimmimg from computer screens and books: The paperless office revisited+, Behaviour and Information Technology, Vol.10, pp257-266, 1991.

[6] Ausubel,D.P. ;The Psychology of Meaningful Verbal Learning. Grune& Stratton, New York, 1963.

[7] K.Hornbaek, E. Forkjaer; Reading Electronic Documents: The Usability of Linear, Fisheys, and Overview+detail Interfaces, In Proceedings of CHI'01, pp293-300, 2001.

[8] T.Hayama, K.Takashi, S.Kunifuji; Document Skimmings Support Environment for Surveying Document in Creative Activities -Towords Supporting Environment for Creative Research Activities, Transactons of the Japanese Society for Artifical Intelligence, Vol19.No2.SP-C, pp.113-125, 2004.

[9] M.Okumura, E.Nanba; The study on the system which automatically summarizes the text. Natural language processing, 6(6), pp.1-26, 1999.

[10] Boguraev,B.K.B,Wong,Y.Y.,Kennedy,C.Bellamy, R.K,E, Brawer,S.and Swartz, J Dynamic Presentation of document content for rapid on line skimming Proc of AAAI Spring Symposium on Intelligent Text Summarisation, 1998.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-108

Torrent worms

Constantinos Patsakis, Nikolaos Alexandris

May 5, 2008

Abstract

One of the most widely used peer-to-peer file sharing protocols is BitTorrent. In this article we focus on the possible use of bittorrent protocol and applications that make use it, as platform for the spread of new computer-worms.

1 Introduction

After the early 70's, when the first computer virus was written, "Cripper", there is constant battle between virus-writers and anti-virus companies that try to protect computer systems. Currently there are more than one million malware products on the Internet. Almost every day there is another break of a new virus. Unfortunately, as the numbers prove it, in most cases, the attackers are one step beyond, thus the need to take precautional measures before the actual outbreak of a virus have to be made.

In the past years there has been has been a tremendous increase in the use of the bittorent protocol. Several reports like [1], show that the peer-to-peer traffic is almost 50-90% percent of all the Internet traffic. Furthermore, a percent of about 50% to 75% percent of this peer-to-peer traffic is using the bittorrent protocol. This means that bittorent is fueling the 25% to 60% percent of internet traffic. At this point we have to point out that a very large portion of this amount of traffic is made from illegal exchange of copyrighted material.

Since the vital part of a computer virus is its spread throughout the Internet, we want to focus on the possibility of a worms using the bittorent protocol for their spread as well other attacks as flooding. The worms that we are going to discuss use the bittorent protocol actively, they are not just copies of a worm that somebody accidentally puts in his torrents. Furthermore, we have to look at the real user of such applications, his computer knowledge and how he reacts to this kind of threats, as well as how are protection software has been designed to cope with this kind of attacks.

2 The protocol and its use

The protocol has been created by Bram Cohen in 2001, [2]. The main advantage of the protocol is that the owner of the original file has to transfer the file once over the net in order to make it available to all clients. The uploader of a file, has to create a small file called torrent, which contains the characteristics of the files to share. This information are the names of files that are shared, the number of pieces that each file is divides, and the hash value of each of these pieces, using SHA-1 hash function. After its creation, the torrent file is uploaded to a server, named tracker which keeps a compete list of all the peers that share this torrent. The file has an "announce" tag that names the tracker that will host this file. Other tags in the protocol, include the name and version of the client used. Each entity that has a complete copy of the original file is called seeder.

The protocol uses the hash function in order to confirm the proper download of each file and is capable of using encryption. On receiving a piece, a peer can seed this piece to other pears. The figure 1, points out how the protocol works.

3 Malicious Software

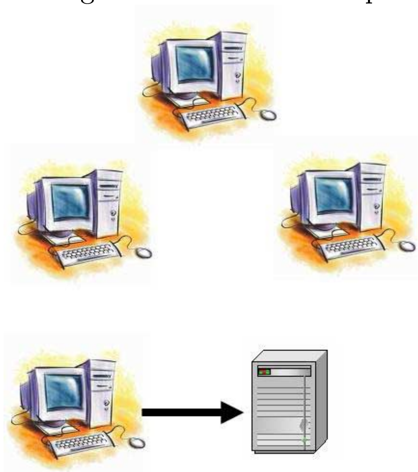
As described in [3] and [4] there are computer viruses that use more complicated approaches than trivial brute force attacks on a network. Nowadays, malicious software can use more dedicated approaches in order to find vulnerable hosts in order to spread themselves. Using search engines like Google and the provided APIs, a computer virus can retrieve information regarding vulnerable hosts.

In the same manner a tracker can be used as search engine for a virus. Firstly, the tracker reports active hosts, thus a virus can have an exact list of true IPs. Secondly, since in most cases a computer is left downloading or seeding without the physical interaction with a user many things can be done during this absence. Meanwhile, as we have said, a very big portion of this traffic is illegal downloads, the users in almost all cases are home users, with little sense of security. A user who is willing to download illegally a film or a software product, in most cases, won't mind running "strange" programs that promise to give access to a certain digit asset. In most cases these systems can be proved to be vulnerable, mostly because of the irresponsible user behavior.

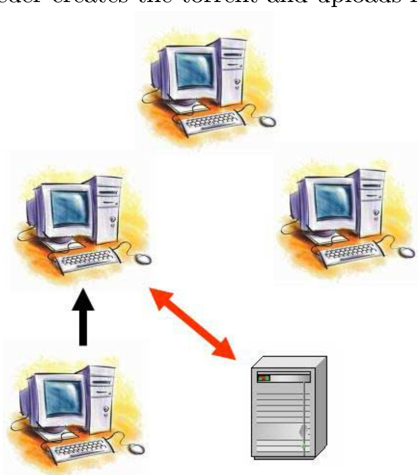
In a case of a virus, the writer appends the virus in the shared content and publishes it on a tracker on the internet, trying to accompany it with an asset that looks "promising" for other users. This can be done almost anonymously, as there are many free trackers and open to public without registrations. The virus is can now be tranfserred to many hosts with their approval. Furthermore the content of the virus can be safely transfered from hosts using the encryption of the protocol, something that many users tend to do in order to hide their traffic from their ISPs. The content, and the virus in advance is for sure to be accessed by the hosts, because this is traffic that each host requested.

The virus can now alter the bittorent client in order to add its viral code

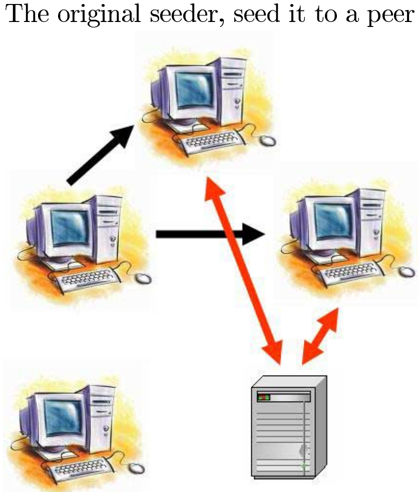
Figure 1: The BitTorrent protocol



The original seeder creates the torrent and uploads it to the tracker



The original seeder, seed it to a peer



Each peer can shares the downloaded pieces

in each created torrent. This means that each host note only continues the growth of the virus by sharing this to other users, furthermore, each time the user creates a new share, the virus is given a new path to vulnerable hosts.

Something that we should note here is that the tracker itself can act as a search engine for the virus. From the announce of each torrent file, the host can find the URL available trackers and download another torrent file, only to get the IPs of active hosts together with the OS platform they are running as well with the version of software. Lists of active hosts lists using can be even be retrieved from temporary internet files, where the torrent files are usually temporarily stored.

4 A case study

In order to see if such a danger is possible we tried to create a "proof of concept virus". The "virus" was written in C++ using Portable Executable code injection to the bittorrent client. The virus upon reboot changes the executable of the torrent client in order to add a dump text file on each created torrent by the user. Of course the growth of the virus now depends on the content that is going to be shared. The tests were made on an Intranet, sharing another empty text file. What we really wanted to see was how will the protection mechanisms will respond to it and if the users will notice the changes.

From the users side, no one would look to his share to see if the shared files had been altered or if his torrent files contain something else that the supposed files. What was interesting to see is that a user would notice that something doesn't goes right, only in the case of a firewall existence. In this case not all firewalls responded the same. We have to note that such a client is not a system file thus most security mechanisms do not pay to much attention to it, yet it is a program that has internet access. Many firewalls asked the user to approve the traffic of this "new" client without prompting that the old one has been altered, others, others judging from the file name granted access.

5 Conclusions

It is obvious that we have to refine our security software, to cope with upcoming threats. The bittorent internet traffic is an at least considerable part of the whole internet traffic, and the users are widely spread all over the globe. If a torrent worm existed that would alter the hash of the recieved pieces, eg altering one bit of a downloaded file, forcing the clients to reload certain pieces. Imagine that this could happen in about 1% of the pieces. The user that shares a torrent will not even notice that something like this happens, yet that would increase the total amount of bittorent traffic by more than 0.5%. Coordination of it can lead to flooding attacks.

On the other hand, the anonymity of peer-to-peer sharing applications in addition to their encryption can be proven to be an extra layer of defence for

the virus. The virus will probably bypass many antivirus programs during its transfer, yet the encryption of the protocol will hide it. The lack of proper handling and proper prompt to the user from usual security software can mislead the user. Finally, we have to note that in many cases such users will bypass many security warnings, as they are used of downloaded files being concidered suspicious from their systems.

References

[1] iPoque Internet Study, http://www.ipoque.com

[2] Cohen, Bram, "BitTorrent Protocol 1.0", http://www.BitTorrent.org

[3] C. Patsakis, N. Alexandris, "New Malicious Agents and SK Virii", International Multi-Conference on Computing in the Global Information Technology, 2007. ICCGI 2007. pp:29 Issue , March 2007

[4] N. Provos, J. McClain, Ke Wang, "Search worms", Conference on Computer and Communications Security archive, Proceedings of the 4th ACM workshop on Recurring malcode table of contents,pp 1-8, 2006

[5] Microsoft Portable Executable and Common Object File Format Specification, http://www.microsoft.com

Software Tools Assisting the Development

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-115

Comparing internal and external software quality measurements

Dimitris STAVRINOUDIS [a] and Michalis XENOS [a,1]
[b] *School of Sciences and Technology, Hellenic Open University, Patras, Greece.*

Abstract. Modern software development companies that have a quality assurance program use measurements and standards to improve product quality as perceived by the users of these products. However, during the entire software life cycle, except for the final customers, different types of 'users' also appear. This paper firstly shows the different views of software quality of these types of users. It also presents the internal and external measurement methods that we used in order to measure the users' opinion of software quality, the benefits and drawbacks of each method, as well as information concerning the techniques used to conduct internal and external measurements. Surveys and examples showing whether software metrics and external views of quality are correlated are also presented. The aim of this paper is to determine up to what point and in which cases can we rely on software metrics in order to define the users' perception of software quality.

Keywords. Software quality, Software metrics, User Satisfaction Measurements

Introduction

In almost all development procedures the end-product quality is strongly connected to measurements that are used to monitor quality factors and to serve as an early indicator of erroneous development. International standards, such as ISO 9001 [1] and the guideline for software development ISO 9000-3, emphasize the need of measurements for assuring product quality. The Capability Maturity Model (CMM) [2] focuses on the need to conduct measurements during software development. As expected, due to their nature, although such standards and models pinpoint the need of software measurements, they do not provide specific guidelines on how to conduct such measurements and what should be aimed in a measurement program. In this paper we present experiences from 10 years of research in the field of software metrics and measurements, discussing how a measurement program is set up, what the goals of such a program should be and how various types of metrics could be used.

Software measurements can be used to prevent catastrophes by offering early warnings of erroneous development, to prevent errors and defects, not only in the early stages of a product's release but even before, and to aid in monitoring software development. A common distinction is to separate measurements into *internal* and *external*, as we discuss further in the following sections. Internal measurements are the ones conducted automatically on program code by the use of internal software metrics, while external are the ones that require the end product and –in many cases– the users'

[1] Corresponding author. 23 Sachtouri Str, GR 26222, Patras, Greece. Tel.: +30 2610 367405; fax: +30 2610 367416. E-mail address: xenos@eap.gr

involvement. While internal measurements can be used to aid in early prevention of errors and defects [3], external measurements can be used to calibrate internal measurement tools and to provide users' perceived measurements of software quality. Although such measurements are subjective, they are accurate indications of users' opinion of software quality. After all, one should never neglect that quality indicates the satisfaction of the end-user needs.

1. Views of software quality

It has been broadly accepted that quality is an abstract concept and cannot be easily understood as a single entity in order to specify measurable elements clearly and directly. Thus, it has been broken down into a number of sub-components or characteristics. These quality characteristics constitute the quality of a product and the overlap among them is the minimum possible. In order to comprehend the meaning of quality and measure it with a systematic method a number of models have been proposed, where the quality characteristics and how they relate to each other are described. An analogous standard that has been proposed and is commonly used is the ISO9126 standard [4]. According to ISO9126, software quality could be broken down into six major characteristics, which are functionality, reliability, usability, efficiency, portability and maintainability and these quality characteristics are split to a number of quality sub-characteristics. The above mentioned product quality characteristics can be divided into two different sets: external and internal. Customers care about external quality characteristics such as: functionality, reliability, usability, efficiency, flexibility, friendliness, simplicity, etc. simply because these are the characteristics which can easily be seen by the use of the product. On the other hand, developers care about internal quality characteristics such as: maintainability, portability, reusability, testability, etc. because these characteristics relate to their development efforts.

However, in a software quality assurance program the term 'user' is not clearly defined. It is assumed that the user is the customer who receives the software from the production company. But there are more types of 'users' during the entire software life cycle. For example, the testing department of a company uses the software before its final release and expects it to meet a set of quality criteria. Similarly, the maintenance department may also be considered as users. Furthermore, companies reuse parts of the code and software components that they had either built previously or procured (obtained) from other companies. In other words, almost everyone that participates in the production plays the role of the user at least at some level.

Despite new methods and techniques being applied to the production process, all such users are still having problems with software of unacceptable quality. Quality is in the eye of the beholder and is judged according to sets of desired characteristics, which hold some significance for the various stakeholders. As a result, different views of software quality may be observed, according to the weight given to each quality characteristic. In other words, there are differences in the perception of factors related to quality as perceived by customers and as perceived by the users within the development team. As it was mentioned earlier, the former are interested more in external characteristics, whereas the latter are interested more in internal ones.

However, similar differences may be observed between users of the same type (e.g. customers), according to how they focus on quality characteristics. For example, one user may emphasise more on efficiency and another more on functionality. The

possible weight given to each of the quality characteristics depends mainly on the type of the software application, how it was produced, for whom and possibly even why. Such questions relate to the users and their requirements. The appearance of different perceptions even of an internal quality characteristic is observed in a development team, according to the specific role of each member of this team. Since the phase during software life cycle that demands the most effort and cost is maintenance (sometimes this cost may exceed the 60% of the total cost of software development) [5], it is interesting to examine the various opinions of maintainability.

2. Measurement methods

As previously mentioned, software quality characteristics can be divided into two different sets: internal and external. Similarly, two different types of measurement methods have been developed according to the aforementioned sets of quality characteristics. In our research we used both internal and external measurements, since these two types are supplementary to each other. In other words, neither of them can completely substitute the other, because of the differences in their goals. In the internal measurements the researcher focuses on the attributes that can be measured purely in terms of the software product, by examining the product on its own, separate from its behaviour. Here, software metrics are used that measure internal product characteristics, without conferring with the users. On the contrary, external measurements are direct measures of external product quality characteristics by questioning the end-users. The researcher focuses on the attributes that can be measured only with respect to how the product relates to its environment. Here, the behaviour of the process is important, rather than the entity itself [6].

What one can obviously derive from the above definitions is that the internal measurements can be applied in a very fast, easy and automated way. Moreover, the implementation of appropriate software tools that can give the results of software metrics is efficient. As a result, the error frequency during this process is minimal and unconsidered and the results of these measurements are considered to be objective and valid. The cost of internal measurements is significantly low, since what is really measured consists of tangible characteristics of the source code of a program. Their results are easily analysed with statistical methods. Furthermore, internal measurements may be applied even before the completion and final release of a product. Consequently, in this case, corrective actions may be followed during the implementation of a software product, before its final release to the customer. However, internal measurements usually do not offer high-level information about the quality of a software product. In other words, the data that derive from them may be difficult to be interpreted and utilised, because of their weakness to be directly related to the external quality characteristics of a software product.

On the contrary, the results derived from external measurements can be interpreted and used directly by the software production company, because they measure straightly the quality of a software product according to these desired external characteristics. External measurements are significant in every software quality program, since all the quality standards emphasize the measurement and the evaluation of the opinion of the end-users. Moreover, the quality data derived from these measurements can be directly analysed, which is a proper fact for software production companies with a high maturity level. Besides, external measurements keep step with the main aim of

software quality and contribute to the improvement of the public image of a company. Finally, the role of the customer during the production phase is promoted, since not only his/her opinion is taken into great consideration, but he/she also participates effectively to the software production process. However, the results from external measurements may be liable to disputes, because they are based on the subjective opinion of users. Here, the error frequency is significantly greater than in the case of internal measurements. In other words, it is difficult to analyse these measurements, due to high error rates and various data scale types. The cost of conducting this kind of survey is very high, because usually it is inefficient to automate it and because human resources in this kind of survey are necessary.

2.1. Internal measurements

Internal measurements can be applied in an automated and cost effective way. Software companies that conduct these measurements use appropriate methods based on software metrics. Numerous software metrics related to software quality assurance have been proposed in the past and are still being proposed. Furthermore, several books presenting such metrics exist such as Fenton and Pfleeger's [6], Shepperd and Ince's [7] and others. As software metrics may be measured automatically, a number of measurement tools have already been implemented. The use of such tools reduces the cost of the measurements and the do not need further inspection for their correctness. In our research we used the software measurement and metrics environment ATHENA [8]. It is a metric-free and language-free metrics environment, which provides completely automated measurements and therefore, the collection of raw data was effortless, as it is required in any enterprise that collects developer-oriented measurements. The selection of the appropriate metrics for internal measurements of one specific product depends on the nature of this product and the programming language used during its implementation. Traditional, broadly used metrics can always be applied regardless of the programming language. The use of internal software metrics is meaningful only when they are applied to large fragments of code. Measurements must not be restricted only to small parts of a software program in order to draw useful conclusions. However, using a framework of metrics allows the programmers to locate modules and routines with a low level of quality. The use of metrics assists the programmers to inspect their code and make the necessary improvements during the implementation phase.

Although in our research we used a large number of metrics, for the presentation of this paper we have selected metrics that can be applied in every routine and every module. In this way, i.e. if we are not restricted by factors such as the specific type of programming languages or programming style or even the type of the application that is being implemented, this paper will be able to lead to generalised and language independent results and conclusions. As a result, we have chosen Halstead's software science metrics [9], McCabe's cyclomatic complexity [10] and Tsai's data structure complexity metrics [11], which are a set of kernel metrics and are representative ones in order to measure size, data complexity and data structure complexity respectively. However, when programming in a specific language or framework of a specific programming style, a software production company must choose an appropriate set of metrics that can also be applied to this style. For example, in object oriented programming, besides the commonly used metrics mentioned above, the developers should also use a set of desired object oriented metrics [12]. Similarly, in web engineering different measurements must also be conducted, as the literature identifies

unique characteristics of web systems that reflect technical, usability and organisational issues [13]. Since in our research different kinds of applications were measured, these specific metrics could not be applied to all of them. As a result, we focused mainly on the commonly used metrics, in order to draw meaningful conclusions.

2.2. External measurements

Most common external evaluation techniques are the heuristic evaluation by experts [14], the performance measurements [15], [16] and the user-perceived software quality measurements [17]. The heuristic evaluation requires a number of 4-5 experts on the specific software type and offers subjective inspection (not measurements). Performance measurements require both experts and users willing to participate at on-site evaluation (performed in a usability lab), thus are expensive and not appropriate in cases of distributed users. Due to all these facts, the external measurements presented in this paper have been conducted using user-perceived quality measurements. This kind of measurements were based on the $QWCO_S$ technique [17] (Qualifications Weighted Customer Opinion with Safeguards technique) that is shown in Eq. (1). $QWCO_S$ technique estimates the result of the external measurements (EM) and is based on the O_i (normalised score of customer i opinion), the E_i (qualifications of customer i) and the use a number of control questions –also called safeguards– where S_T is the total number of control questions, from which the customer i (from a total of n customers) has replied correctly at S_i.

$$EM = QWCO_S = \sum_{i=1}^{n}\left(O_i \cdot E_i \cdot \frac{S_i}{S_T}\right) \Bigg/ \sum_{i=1}^{n}\left(E_i \cdot \frac{S_i}{S_T}\right) \qquad (1)$$

Although such external measurements cannot be automated as internal measurements are, effort can be made on the automation of the data collection, the testing of the validity of the responses (using the aforementioned technique) and the analysis of the results (that is a typical statistic analysis). For this purpose the *CESM*[2] Tool [18] has been used. *CESM* Tool allows the use of a measurements technique. The *CESM* is comprised of the Questionnaire Designer, the surveys web pages, the Questionnaire Analyser and a database. The author of the questionnaire has to specify which user–oriented quality characteristics derived from the (included in the tool) ISO9126 classification will be dealt with. The questions of the questionnaire must be clustered in groups, according to which quality characteristic they refer to. Each group of questions can be given a different weight, depending on the emphasis given to the corresponding quality characteristic by the author.

3. Results from the surveys

Modern software developers that have a quality assurance program use metrics and standards to improve product quality as perceived by the end-users of the product. After all, the users' view of quality is usually considered to be paramount. According

[2] *CESM* is an acronym for: 'Collecting External Software Measurements'.

to Fenton's axiom [6], good internal structure is expected to provide good external quality. However, the main question that arises is up to what point and in which cases may a software production company rely on the use of metrics in order to guarantee a priori a satisfying score in the external measurements of its products and, in other words, acceptance by the users. Furthermore, another research question is when and under which circumstances a low score in these metrics indicates a possible failure of the software product's quality, as this will be perceived by the users. Our basic aim was to identify cases that users accept as having high quality, with early-measurable internal metrics as potential indicators. In the following sections two different surveys are presented. The survey in section 3.1 examines the correlation between internal measurements, which were conducted as described in section 2.1, and external measurements that indicate the end-users' opinion of quality. The survey in section 3.2 examines the correlation between similar internal measurements of software products and external measurements of the opinion of programmers which would have to maintain or reuse components of these products.

3.1. End-user survey

This presented survey was based on end-users and included data from $N=34$ components of software products developed for various projects. Each of these N_i $(i \in [1, 34])$ components could be measured as a stand-alone product, therefore it could be evaluated separately to other components. All these components were assessed using external measurements. The results presented in this section are based on external measurements from K_i users for each component N_i. The number K_i varied for each component N_i and applies that $K_i \in [35, 56]$. End-user survey data were combined in a single measurement result for each component, using the Eq. (1), thus resulting a single number that combines the external measurements for component N_i. Due to the nature of the result of the Eq. (1), which hereafter will be represented as EM_i for the component i, it applies that $EM_i \in [0, 1]$.

Since we found a very high correlation between the results of the internal metrics that we used, we focused on the *language level* $\lambda=N(\log_2 n)(2n_2/n_1N_2)^2$, which measures how well the programming language is used, on the *essential size ratio* $R=(n_1\log_2 n_1+n_2\log_2 n_2)/N$, which detects if the code contains any code impurities (where n_1, n_2 is the number of distinct operators and operands, N_1, N_2 the number of total operators and operands, $N=N_1+N_2$ is the program size and $n=n_1+n_2$ is the vocabulary), on the *cyclomatic complexity* $V(g)=e-n+2$ proposed by McCabe (where e is the number of edges and n is the number of nodes) and on *data structure complexity* proposed by Tsai. Laying a greater weight on larger routines, the language level of the whole component can be defined as $\lambda=(\Sigma N_i\lambda_i)/\Sigma N_i$ (where N_i is the program size and λ_i is the language level of the routine i). Equivalently for cyclomatic complexity the factor $10/V(g)$ was used, where 10 is the proposed by McCabe maximum complexity, and the measure of the cyclomatic complexity of the whole component can be defined as $V(g)=(\Sigma N_iV(g)_i)/(\Sigma N_i)$ (where $V(g)_i$ is the cyclomatic complexity of the routine i). Finally, from data structure complexity the factor $1/(T+1)$ was used, where T is the greatest degree of the polynomials of the component's routines that derive from the Tsai's formula $C(K)=\Sigma(C(K_i))+S(K)$ and $S(K)=(v+e)x^L$ (where v is the number of nodes, e is the number of edges and L is the number of simple circular paths into the reduced data graph). In the derived polynomials, as the degree of a polynomial grows,

the data structure complexity of the program also grows. So the factor 1/(T+1) is disproportional to data structure complexity. In order to summarize all the above mentioned metrics, a combined formula IM_i for each component i was formed, which is shown in Eq. (2). It must be made clear that the above formula does not measure a physical quantity of the component, but is both a customized collective formula created especially for this particular case study and a means by which the results of all the metrics can be combined. The purpose of IM is to facilitate the presentation of the results providing an estimation for the overall performance of each component regarding to the metrics used.

$$IM = 0.2\lambda + 0.2R + 0.4\left(\frac{10}{V(g)}\right) + 0.2\left(\frac{1}{T+1}\right) \quad (2)$$

Both external measurement data (Normality test Shapiro-Wilk's Significance level = 0.857) and internal measurement data (Normality test Shapiro-Wilk's Significance level = 0.901) were distributed normally. Correlations among the internal metric results and the external measurements were found to be equal to 0.819 which was statistically significant. The scatter plot shown in Figure 1 illustrates this measured correlation. The external measurements are shown on the horizontal bar, whereas the internal measurements are shown on the vertical bar. The line on the scatter plot shows where all points should be in case these two variables were 100% correlated. It is obvious that components with low score in the internal metrics will also have low score in external measurements. Besides there is no measured component, which has low score in the internal metrics while having high, or even average, score in the external measurements. On the contrary, high internal metric results do not necessarily imply high external measurements results. As the scatter plot shows, there are many components that score high in internal metrics, but score poorly in external measurements.

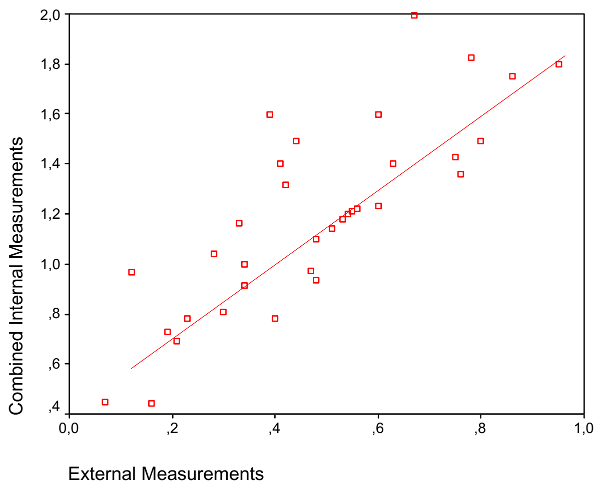


Figure 1. Correlation between IM_i and EM_i

Therefore, the derivative of this survey is that poor internal metric results in most cases would result equally poor results in the external measurements. Consequently, it

is generally acceptable to set a limit to carefully selected internal metrics in order to preserve the external quality of a component as perceived by its users. But, there is no reliable method to find out if the components with high internal metrics score will also have an equally high score in external measurements. Internal metrics can serve as a good first indication for user perceived quality, but they cannot always guarantee successful external measurements as well.

3.2. Inside the company survey

Software maintainability is a difficult factor to quantify. However, it can be measured indirectly by considering measures of design structure and software metrics. It is also claimed that logical complexity and program structure have a strong correlation to the maintainability of the resultant software [19]. This survey presents the relation between software metrics and maintainability and was based on the opinion of future maintainers. The survey included data from *N=29* components of software products developed for two large software projects. Similarly with the previous survey, each of these N_i $(i \in [1, 29])$ components could be measured as a stand-alone product, therefore it could be evaluated separately to other components. During the software life cycle these components had to be maintained or even modified in order to be reused in future projects. The results presented in this section are based on external measurements from P_i programmers for each component N_i. The number P_i for each component N_i varied from 28 to 54. The opinions of these programmers were combined in a single measurement result for each component, using the Eq. (1), thus resulting a single number EM_i that combines the external measurements for component N_i. It is obvious that, as mentioned in the previous survey, that $EM_i \in [0, 1]$. In this survey 4 outliers were observed and weren't taken into consideration, since they consisted of either declaration routines or very short routines without any control flow statements that were simply calling other routines.

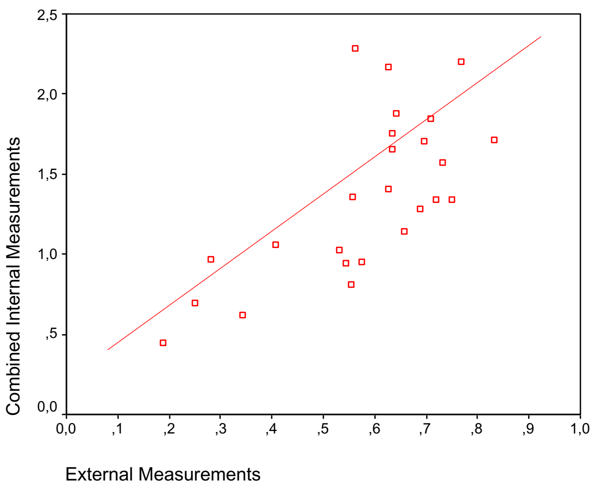


Figure 2. Correlation between IM_i and EM_i

In a similar manner to the external measurements, the results from the internal metrics were analysed and compared to the external measurements. Moreover, the

same combined measure IM_i was used. Both external measurement data (Normality test Shapiro-Wilk's Significance level = 0.439) and internal measurement data (Normality test Shapiro-Wilk's Significance level = 0.732) were distributed normally. Correlations among the internal metric results and the external measurements were found to be equal to 0.730 which was statistically significant. The scatter plot shown in Figure 2 illustrates this measured correlation.

From the results of this survey it was obvious that software metrics' score and programmers' opinion of maintainability were highly correlated. In detail, components with high internal measurements scored equally high in external measurements. Additionally, components that failed in software metrics scored poor according to the programmers' opinion of maintainability.

Furthermore, it was also observed that some components that failed in metrics scored with a great variation according to the programmers' opinion of maintainability. In other words, low internal measurements do not necessarily imply low external measurements. This variation was observed mainly because of the nature of the application these modules belong to. This observation is in conflict with the relation between software metrics and the customers' opinion of quality, as it was shown in the survey of section 3.1. In the case of customers, metrics can only detect possible causes for low product quality, whereas satisfaction of internal quality standards does not guarantee success in fulfilling the customers' demand on quality. On the contrary, in the case of the maintenance phase, good internal structure implies high score of maintainability as this will be perceived by the programmers, whereas low internal measurements do not necessarily entail low score of maintainability. Therefore, the derivative of this survey is that poor internal metric results in most cases would result equally poor results in the external measurements. Consequently, it is generally acceptable to set a limit to carefully selected internal metrics in order to preserve the external quality of a component as perceived by its users.

4. Conclusions

Software metrics are able to offer a very good first indication for external quality. It was shown in both presented surveys that they are highly correlated with external quality characteristics. Specifically, they provide an easy and inexpensive way to detect and correct possible causes for low product quality as this will be perceived by the customers. Setting up measurement programs and metric standards will help in preventing failures to satisfy customers' demand for quality. Programmers should keep the metrics score of their programs within acceptable limits. This can be accomplished by the use of appropriate programming environments that offer basic quality assurance functions, such as setting a source code measuring program, and define basic metrics and their limits. These environments may also offer comparisons between different versions of the same project, allowing the programmers to monitor the progress of their performance in relation with the measurement goals they have set or with the requirements of the quality assurance program being followed.

Generally, the better results the metrics achieve when measuring software modules, the higher level of maintainability according to programmers opinion these modules will have, so the less effort will be spent during the maintenance process. This conclusion was also observed when measuring different versions of the same component, where the subsequent ones achieved better results in internal measurements

than their former. Consequently, sometimes it is preferable for a software development company to risk not only to correct, but also to rewrite specific parts of working code in order to decrease the effort during the maintenance phase and to avoid or early cure possible customers' problems. However, if a component fails in internal measurements it is preferable not to reject it automatically although it usually fails in external measurements, too. On the contrary, as low score in metrics sometimes is expected, because of the nature of what a component may implement, the programmers should inspect carefully their code and decide whether it needs to be corrected (or even re-written) or not.

Acknowledgement

The work presented in this paper for the last 4 years is funded by the project: "Support of Computer Science Studies at the Hellenic Open University" (2003-2008, Greek Ministry of Education: Operational Programme "Education and Initial Vocational Training").

References

[1] ISO/IEC 9001:2000. Quality management systems - Requirements, International Organization for Standardization, 2000.
[2] Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V., 1993. Capability Maturity Model for Software (Version 1.1), Pittsburgh: Carnegie Mellon University - Software Engineering Institute: Technical Report. CMU/SEI-93-TR-024.
[3] Shepperd, M.J., 1993. Software Engineering Metrics, Volume I McGraw-Hill, Maidenhead,England.
[4] ISO/IEC 9126. Software Product Evaluation – Quality Characteristics and Guidelines for the User, International Organization for Standardization, Geneva, 2001.
[5] Bell, D., Morrey, I., Pugh, J., 1992. Software Engineering: A Programming Approach, 2nd ed., Prentice Hall, isbn:0-13-832536-7.
[6] Fenton, N., Pfleeger, S., 1997 Software Metrics A Rigorous & Practical Approach, Thomson Computer Press.
[7] Shepperd, M.J., Ince, D., 1993. Derivation and Validation of Software Metrics, Clarendon Press, Oxford, UK.
[8] Tsalidis, C., Christodoulakis, D., Maritsas, D., 1991. Athena: A Software Measurement and Metrics Environment, Software Maintenance: Research and Practice.
[9] Halstead, M.H., 1975. Elements of Software Science, Elsevier Publications, N-Holland.
[10] McCabe, J., 1976. A complexity measure, IEEE Transactions of Software Engineering, SE-2(4).
[11] Tsai, T., Lopez, A., Rodreguez, V., Volovik, D., 1986. An Approach to Measuring Data Structure Complexity, COMPSAC86, pp. 240-246.
[12] Xenos, M., Stavrinoudis, D., Zikouli, K., Christodoulakis, D., 2000. Object-Oriented Metrics - A Survey, FESMA 2000, Federation of European Software Measurement Associations, Madrid, Spain.
[13] Overmyer, S., 2000. What's Different about Requirements Engineering for Web Sites?, Requirements Engineering Journal, 5 (1), pp. 62-65.
[14] Nielsen, J., Mack, R.L., 1994. Usability Inspection Methods, Wiley.
[15] Dumas, J., Redish, J., 1993. A Practical Guide to Usability Testing, Ablex, Norwood, New Jersey.
[16] Rubin, J., 1994. Handbook of Usability Testing, John Wiley and sons, New York.
[17] Xenos, M., Christodoulakis, D., 1997. Measuring Perceived Software Quality, Information and Software Technology Journal, Vol. 39, pp. 417-424.
[18] Xenos, M., Stavrinoudis, D., Christodoulakis, D., 1999. CESM: a Perceived Software Quality Assessment Tool, 4th International Conference on Software Process Improvement - Research into Education and Training, INSPIRE '99, Heraklion, Crete, Greece, pp. 155-164.
[19] Kafura, D., Reddy, R., 1987. The Use of Software Complexity Metrics in Software Maintenance, IEEE Trans. Software Engineering, vol. SE-13, no. 3, pp. 335-343.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-125

An Assertion Mechanism for Software Unit Testing to Remain Unaffected by Program Modification

The Mechanism to Eliminate Dependency from/to Unnecessary Object

Masanori SAKAKIBARA [a,1], Kohei SAKURAI [b] and Seiichi KOMIYA [a]
[a] *Graduate School of Engineering, Shibaura Institute of Technology*
[b] *Graduate School of Arts and Sciences, the University of Tokyo*

Abstract. Assertions are used for unit testing in Object-Oriented Programming. With assertions, a test programmer compares with an expected value the result of a test execution which is obtained by referring to the object containing its result in the object being tested. Therefore, there is a problem that test classes described as testing code depend strongly on the unnecessary objects which are not of concern to unit testing. In this work we developed an assertion mechanism which traverses objects automatically in order to eliminate the dependency from/to such unnecessary objects. Furthermore, we performed experiments on open source products, and confirmed that using this assertion mechanism decreases the coupling between a test class and other classes.

Keywords. Unit testing, assertion, Object-Oriented Programming

Introduction

In the software development using object oriented programming (OOP), unit testing ensure the software reliability. In unit testing, a test programmer write multiple test cases for each module with which the program being tested is configured. Therefore, running a test case with a test driver, unit testing are performed.

In the software development using OOP language, like Java, the JUnit testing framework [6] are often used as a test driver. With JUnit the test case is described as a class (hereafter called a test class), and running test cases is automated.

A test class targeted a certain class and its methods for testing and mainly the following description is made. First, the specific test input for the method of a test, and the object of the class for the test is supplied. Next the method to be tested are executed with the supplied input.

[1]Shibaura Institute of Technology toyosu 3-7-5, etoku, Tokyo, 135-8548 Japan; E-mail: m106045@shibaura-it.ac.jp

Finally the output of the method or the values of changed state are compared with the expected results using assertions. If the comparison check of the assertion fails, it means there are some defects in the method to be tested. The value used for comparison of assertion are obtained through reference of other related objects from the object of the class being tested.

In OOP, by changing the program being tested by refactoring etc. the definition of the class or the method of the class, not being tested, get changed in many cases. As a result, the test programmer has to change the test class manually. Such changes need to be avoided by improving the robustness of the test classes related to the changes in the programs being tested.

In this work, regarding the unit test, from the objects being tested at the start to the value (executed results) of the objects, which can be reached, when comparing with assertions, the problem of the robustness of changes to the programs is discussed. The objects of the class not to be tested, are called unnecessary objects. The dependency between the object being tested and such unnecessary objects is pointed as decreasing the robustness of the test class.

And in order to eliminate the dependency between unnecessary objects, we propose *TraversalAssert*, a new assertion mechanism, developed with the mechanism Demeter [4], which uses traversal. The Demeter has the functions to traverse the graph (object graph) by the reference relation of an object, indirectly related to another object, and obtaining its relation references.

In order to use the traversal easily in unit testing, the authors have developed an assertion mechanism as a library, to traverse the object graph on the basis of the object of the class for testing.

In this paper, this assertion mechanism is applied to the actual open source test program, and it explains that the dependency between unnecessary objects can be eliminated from a test class.

Furthermore, the verification experiment was conducted for the several open source software products, and quantitative evaluation was performed using the degree of coupling between the classes [3], which are well known software matrics.

The configuration after this paper is, as follows. Section 1, explains the problems regarding the robustness of unit test by a concrete example. Section 2, explains about an assertion mechanism to be proposed. Section 3, explains the implementation of the proposed assertion mechanism. Section 4, explains the validity of the proposed assertion mechanism by experiment and check of the results. Section 5, explains about the related work. Section 6, gives the conclusions and the future work.

1. Motivation Example

In this section, the problem of the dependency between the unnecessary objects in the test class for unit testing are explained by using an actual example. Commons-Betwixt[2] currently being developed by Jakarta Apache Project as open source software is taken up. Commons-Betwixt is a library to inter-relate Java beans and XML document.

Henceforth, in Section 1.1, an actual test class is taken from Common-Betwixt and explained. In Section 1.2, problems are explained.

[2] `http://commons.apache.org/betwixt/`

```
1  public class TestXMLBeanInfoDigester ...
2    public void testDigester() {
3     XMLBeanInfoDigester digester =...
4     InputStream in =...
5     XMLBeanInfo info=
6       (XMLBeanInfo) digester.parse(in);
7     ElementDescriptor descriptor =
8       info.getElementDescriptor();
9      .../*check null of descriptor*/
10    ElementDescriptor[] elements =
11      descriptor.getElementDescriptors();
12     .../*check null of elements*/
13    assertEquals("channel",
14      elements[0].getLocalName());
15   }
16 }
```

Figure 1. A sample test class.

1.1. *Example: Test Class for Syntax Analysis Program*

In Fig. 1, `TestXMLBeanInfoDigester` class is taken in parts from the definition of a test class for `XMLBeanInfoDigester` to be tested. In general, unit testing is performed by executing such test classes after the class `XMLBeanInfoDigester` and other related classes are implemented.

The method `testDigester` is targeted for testing syntax analysis of XML document with the method `parse`. Actually, the object of class `XMLBeanInfoDigester` and the parameters of class `InputStream` are instantiated and set up (lines 3-4). After calling the method `parse` with parameter `in` together (line 6), the returned value of the method is checked (lines 7-14).

In order to check the returned value of the method `parse`, the test programmer gives the expected value which serves as the element name in an XML document (in Fig. 1, line 13 `"channel"`) to the test class, and the value has to be compared with the actual element name, called by the assert statement (`assertEquals`)(lines 13-14). In order to get the actual value, from the object of type `XMLBeanInfo` (lines 7-8), the object of type `ElementDescriptor` (lines 10-11) is acquired in order.

1.2. *The Dependency on Unnecessary Objects*

The code which calls methods one after the another in an object which has a reference relation with a certain object is called Law of Demeter(LoD)[5], and is against the style of calling program. Here, the reference relation of an object calls the method from a certain object, and by referring to the fields or referring to the table having those results can acquire the references, and points to the relation with other objects. The programs which do not obey LoD, have a difficult structure and to maintain.

LoD must use only the module of the limited object closely related to the module. Although LoD is the rules of a programming style in developing software program, we consider applying the definition of LoD to the test class. The limited object which is

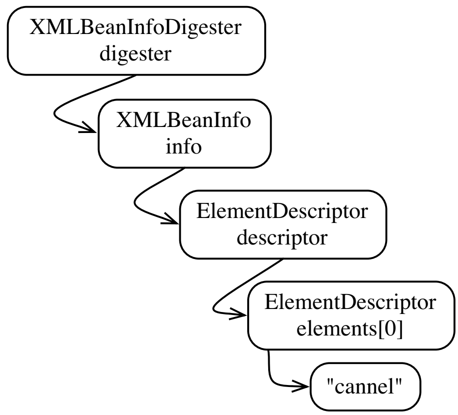


Figure 2. Partial object graph.

closely connected with a test class, 1) the object for test, 2) the input object for the test, [3] and 3) the object of the values for the check of output results or state of changes used in assertion. The class and method of these object groups can be called the limited modules, which are closely related to test class. All other objects, which have reference relation to the test related objects used in test class are unnecessary. Unnecessary object is used mainly in order to acquire the actual value of the result to compare with expected value.

Fig. 1 checks the return value of method by sending method to the object of `XMLBeanInfo` and `ElementDescriptor` in order, including the violation code of LoD. Fig. 1 is the partial object graph made from the reference relation of the object when the code of Fig. 1 is executed. The ellipses in the figure show the objects and the arrows show the reference relation between the objects.

- The top object of the Fig. 2 shows a object stored in the variable `digester` of `XMLBeanInfoDigester`, which is the target class for testing in Fig. 1.
- The `"channel"` at the bottom of Fig. 2 is the object of type String, which should be checked in the test. It is referred when the method of line 14 is called in Fig. 1.
- In the middle of Fig. 2, 3 objects of types `XMLBeanInfo` and `ElementDescriptor` are said as the unnecessary objects, which causes violation of LoD. These unnecessary objects are referred in lines 7-13 of Fig. 1. The objects stored in the variable `info` of type `XMLBeanInfo`, second from top, are referred as the result of when method `parse` is called for top ranking objects.

The robustness of a test class is decreased by LoD violation of a test class, because the presence of the unnecessary objects, which have reference relation from the object of the class for a test, can receive the changes effects of the unnecessary objects. For example, when the name of the method of the class `ElementDescriptor`, which is the type of the unnecessary objects used in Fig. 1, is changed the codes of the test class also need to coincide with the name. Moreover, when the definitions of class

[3]The classes of objects for inputs are prepared for each test class by using a stub or Mock Object etc. So, it can be said that they are closely connected with a test class.

```
public class TestXMLBeanInfoDigester ...
  public void testDigester() {
   XMLBeanInfoDigester digester =...
   InputStream in =...
   from(digester.parse(in))
     .to("channel").verify();
  }
}
```

Figure 3. Test class rewritten with TraversalAssert.

`ElementDescriptor` are deleted, a test programmer have to rewrite the procedure, which acquires the actual element name.

2. Assertion Mechanism for Elminating The Dependency on Unnecessary Objects

By using the mechanism of *Demeter*[4] for traversal the problems of violation of LoD regarding the test class, explained in Section 1.2, is resolved by developing assertion mechanism for unit testing.

First in Section 2.1, below, the explanation of Demeter to be used in the development of assertion mechanism is given. Then Section 2.2 outlines an example of use of assertion mechanism, and advantages.

2.1. Demeter: Mechanism for Traverse

Demeter is the mechanism proposed in order to realize LoD. By automatically traversing the objects graph with Demeter, it becomes unneeded to acquire reference of objects other than the objects proposed at the top of object graph. As a result, codes, which call methods one after another for the objects in the reference relation are not required, and LoD can be realized.

Demeter traverses the graphs of reference relation of the starting objects by the given traversal strategy, and executes procedure to the reached object. Traversal strategy is the requirement of the path to object graph. The strings about the type of objects are defined by declaration of valid expressions. For example, about the object graph at the top of Fig. 2, to arrive at the end object `"channel"`, the path to object `String`, which is obtained by the reference relation of from the starting object `XMLBeanInfoDigester` to the object `XMLBeanInfo` and 2 objects `ElementDescriptor`, is given as traversal strategy.

DJ[8] is developed as a processor of Demeter. DJ can be used as a library of a Java language without any extension of the language. DJ describes literal a character string of Java as traversal strategy. The visitor object is given as the executing procedure for the reached object.

2.2. Outlines of Assertion Mechanism

The Assertion library *TraversalAssert* for testing was developed in order to describe Demeter by assert statement, without extending Java language. In order to generalize

the processing of the existing Demeter, such as DJ, when used in test classes includes many lengthy or duplicate descriptions in the traversal strategy or the procedures. The developed assertion mechanism applied beforehand the traversal strategy and procedure, which are often used to check of the values of test class results, as application programming interface (API).

TraversalAssert as described in Section 2.1, is if the comparison of actual results and expected values is prepared for test input, it can be used any time for the usual assert statement assertEquals and also for the test related class by using API.

Fig. 3 is the test class, rewritten by the foregoing example using the TraversalAssert library. In lines 5-6, it explains that from the object `XMLBeanInfoDigester` it can reach to the `String "channel"`.

When a test class of Fig. 3 is executed, In TraversalAssert the object graph included Fig. 2 is automatically traversed by the visitor in the assertion mechanism. If the object of the type `String "channel"`, which is the target object, can be reached, the assertion is satisfied and the test becomes successful. If it is not reached, the test fails and an error message is outputted, as follows.

```
not XMLBeanInfoDigester -> "channel"
```

The test class using TraversalAssert does not require the description of acquiring the reference of unnecessary objects manually, their dependency is not required and LoD is realized. In Fig. 3, there is no need to describe the codes acquired by reference of `XMLBeanInfo` objects and `ElementDescriptor` objects from `XMLBeanInfoDigester` objects by comparing with Fig. 1.

Therefore, it is hard to get the changes effects of the classes of `XMLBeanInfo` or `ElementDescriptor` and so increase the robustness regarding changes. For example, even if the method name of `ElementDescriptor` changes, or class definitions are deleted, by implementing the test class without re-writing, the unit test can be done.

3. Implementation of Assertion Mechanism

This section briefly explains an implementation of TraversalAssert. The authors implemented a prototype of TraversalAssert as a Pure Java library by using DJ[8], which realizes LoD through traversal, with about 1100 lines of code.

In DJ the class graph, expressed by the class relation of the class in the specified package is generated. For example, by loading the code of Fig. 3, the class graph shown in Fig. 4 is generated. Regarding the class graph, which class is to be used as traversal source, and through which class, and to traverse till which class, defined traversal strategy can be specified.

In TraversalAssert, by using traversal strategy and the object visitor to perform traversal, successful functional achievement of assertion mechanism is realized.

In order to check that performance, the time required to carry the test regarding 19 test classes of Commons-Betwixt, is measured. The measured environments were with Intel CPU, Celeron 3.2GHz, memory 512MB, Linux (kernel 2.6.9-42) has JavaVM version 1.5.0. As a result, according to TraversalAssert, required more than 200 seconds after rewriting as compared to 2 seconds before rewriting.

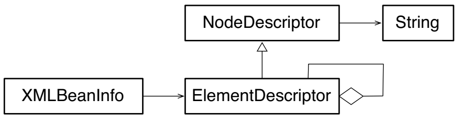


Figure 4. Partial class graph.

The time difference before and after rewriting is the time to traverse automatically the graph of the objects of origin for testing. In other words, before rewriting, from the objects being tested through the related objects, the results of a test execution obtained manually. And after rewriting, at the time of test execution, by traversing the results from the object being tested, it was found that the manual job is done automatically. In future, it is necessary to optimize the efficiency of traversal.

4. Verification Experiment

In order to examine the validity of developed TraversalAssert, the verification experiment by comparison of assert statement of JUint and the case when TraversalAssert is applied, was done.

The program for the experiment selected the software product of Jakarta project Commons-Betwixt[4], Commons-Digester[5], and Commons-Scxml[6], HttpCore[7]. A test class exists for these products.

In order to evaluate, Coupling between Object Classes (CBO) [3], which is one of the matrices of CK is used. CBO is the number of other classes which a class is coupled to. The coupling here means, a certain class is referring to the methods or argument instances of other class. The class of high CBO value depends on many other classes, and it is easy to get influenced by changes. So the robustness of the test class is lowered.

By experiment, CBO value of the test classes existing in each product and the rewritten test class by TraversalAssert, was measured. But the classes included in the standard library, arranged type objects, primitive types, and the classes in the JUnit and TraversalAssert library with no changes are not counted as the classes included in CBO value. Further the test classes as they are not counted, but the internal classes are counted.

The result of measurement is summarized as follows.

- Table 1 shows the total number of test classes for each product, and test class number with which CBO value was lowered after rewriting.
- Table 2 shows the total number of packages for each product, and the packages number, including the test classes with which CBO value was lowered after rewriting.
- Table 3 shows the low, high, average comparative ratio of CBO value before rewriting, and lower value after rewriting.

[4]`http://commons.apache.org/betwixt/`
[5]`http://commons.apache.org/digester/`
[6]`http://commons.apache.org/scxml/`
[7]`http://jakarta.apache.org/httpcomponents/httpcomponents-core/`

Table 1. The number of test class with lowered CBO value.

Product	Total	Lowered number	Ratio
Commons-Betxixt	103	19	0.18
Commons-Digester	32	7	0.22
Commons-Scxml	32	15	0.47
HttpCore	61	9	0.15

Table 2. The number of package which include test class with lowered CBO value.

Product	Total	Included number	Ratio
Commons-Betxixt	19	9	0.47
Commons-Digester	4	2	0.50
Commons-Scxml	10	5	0.50
HttpCore	12	7	0.58

Table 3. Lowered rate of CBO value.

Product	Min	Max	Average
Commons-Betxixt	0.077	0.667	0.240
Commons-Digester	0.143	0.250	0.179
Commons-Scxml	0.100	0.500	0.262
HttpCore	0.083	0.500	0.219

With TraversalAssert library 20 percent test classes of the total test classes the value of CBO was revised, lowered. The packages containing the test classes, which revised (lowered) CBO value, are about 50 percent of the total packages, and were greater than the ratio of test classes. This shows that the dependency of test classes between unnecessary objects are scattered over many packages. If the packages design is based on the functional unit of products, not only for a special package, but TraversalAssert library can be used for various packages.

Moreover, the number of dependent classes of the unnecessary objects, were able to be eliminated about 1-3 times. Because of this, CBO value of test classes, were able to be lowered by 20 percent as a result. The maximum test classes of the ratio, which lowered CBO value, the number of objects required for the setup regarding the test was low. On the other hand, the minimum test classes of the ratio, which lowered CBO value, the number of objects required for the setup regarding the testing was high. How much the TraversalAssert library can lower the value of CBO depends upon the number of necessary objects required for setup of testing.

5. Related Work

In unit testing, to guarantee the quality for each module, the technology and the problems for are described.

5.1. Mock object

Mock Objects [9] are the techniques to support the test-drive development [1] of object-oriented programs. In the test- drive development, the development process is advanced

in steps through unit tests. Mock objects play the role of imitating the functions of the objects which the object being tested collaborates with. Objects which the object being tested are actually related to are replaced with mock objects, So the objects being tested do not depend on other related objects. As a result, the test programs are structured highly, and the test related programs preserve the encapsulation and maintainability is improved.

However, mock objects regarding the objects being tested are arranged for input of each test class, and have close connection with a test class. Therefore, by refactoring etc. if mock objects are changed then test class must also be updated. By using the mechanism of traversal of the graph of an object dynamically, TraversalAssert eliminate the dependency on mock objects and improves the robustness of the test class.

5.2. Design by Contract

The method of guaranteeing the quality of an object-oriented program there is Design by Contract (DbC) [7] proposed by Meyer. In DbC, the general conditions, which a program should fulfill for every routine as a method, are described as assertion. By using the assertion although it is possible to develop the software which are reliable by pinpointing defective parts, but it is not easy to describe the general conditions to fulfill.

In TraversalAssert of this work, there are not general conditions for main part of the program and the separated test class, but more specific conditions are described as assertion. The actual expected values regarding the execution result of a certain input, by comparing in the form of the actual execution result being described as conditions of the test, the description of assertion is very clear.

Moreover, by the module specifications describing the assertion of DbC as per (Java Modeling Language) JML, there are arrangements to generate the test classes automatically [2]. If the test classes can be generated automatically, it becomes unnecessary for a test programmer to describe a lot of test classes, and reduce the labors of update of the test classes by changes in a test. But the general specifications using JML becomes too much descriptions, and become weak to changes of modules for the test and become necessary to update the description of the specifications. Moreover knowledge of specifications description is also required.

In the TraversalAssert library, regarding the executions results of modules for unit testing and the input data to these modules only the actual expected values are described. For that reason, the description become brief and different from the specifications by JML, and become strong to the changes of the modules for the test.

6. Conclusion

In this paper, in the software development by OOP language, the dependency between unnecessary objects of unit test, the problems of loosening the strictness of test classes are shown.

As a solution of the problem, by using the mechanism Demeter for traversal, assertion mechanism TraversalAssert was developed to elminate the dependency between unnecessary objects.

TraversalAssert by using DJ processor of Demeter implemented the easy use of traversal in unit tests.

In order to verify the validity of developed TraversalAssert, a verification experiment is conducted for a number of software, for which the test classes are announced as open source, and improvement in robustness of test classes was checked.

In future, for reducing the time required for automatic traversal by TraversalAssert, the optimization of traversal strategy, improvement of robustness, specifying the execution results for a test, and the side effects of the changes to the state of the objects by traversing stack, etc. are the subjects.

References

[1] Beck, K.: *Test Driven Development: By Example*, Addison-Wesley Professional; 1st edition, November 8, 2002.

[2] Cheon, Y. and Leavens, G. T.: A Simple and Practical Approach to Unit Testing:The JML and JUnit Way, *the European Conference on Object-Oriented Programming (ECOOP)*, Iowa 50011-1040, USA, Springer Verlag, June 10-14, 2002.

[3] Chidamber, S. R. and Kemerer, C. F.: A Metrics Suite for Object Oriented Design, *IEEE Traansactions on Software Engineering, VOL. 20, NO.6* , June 1994.

[4] Lieberherr, K. J.: *Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns*, PWS Publishing Company, Boston, 1996. ISBN 0-534-94602-X.

[5] Lieberherr, K. J., Holland, I., and Riel, A. J.: Object-oriented programming: An objective sense of style, No. 11, San Diego, CA, September 1988, pp. 323–334. A short version of this paper appears in *IEEE Computer Magazine*, June 1988, Open Channel section, pages 78-79.

[6] Louridas., P.: Junit: Unit testing and coding in tandem, *IEEE Software, 22(4):12 - 15*, July-Aug 2005.

[7] Meyer, B.: Applying Design by Contract, *IEEE Computer, 25(10):40-51*, October 1992.

[8] Orleans, D. and Lieberherr, K.: DJ: Dynamic Adaptive Programming in Java, *Reflection 2001: Meta-level Architectures and Separation of Crosscutting Concerns* , Kyoto, Japan, Springer Verlag, September 2001. 8 pages.

[9] Steve Freeman, Tim Mackinnon, N. P. J. W.: Mock Roles, Not Objects, *Proceedings of the ACM SIGPLAN 2004 Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Berkshire House, 168-173 High Holborn, ACM Press, OOPSLA2004.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-135

Accessibility Evaluation for GUI Software Using Source Programs

Junko SHIROGANE [a] Takashi MORI [b] Hajime IWATA [c] and Yoshiaki FUKAZAWA [b]

[a] *Tokyo Woman's Christian University*
[b] *Waseda University*
[c] *Kanagawa Institute of Technology*

Abstract. Computers and software are now widely used in human society. With their dissemination, it becomes necessary for disabled and elderly people to use software. However, it is difficult to develop accessible software for them. Also, there are many guidelines and support tools for developing accessible web sites. However, for software, such guidelines and support tools are few. In our research, to develop accessible software easily, we propose a method of evaluating the accessibility of Graphical User Interface (GUI) software. In our method, source programs of GUI software are analyzed, the accessibility of the GUI software is evaluated along with the accessibility guidelines, and the list of unsatisfactory codes for accessibility and indications on how to modify them are shown.

Keywords. GUI, Accessibility, Source programs

1. Introduction

It has been become important for disabled and elderly people, as well as physically unimpaired people, to use computers widely. However, software is not always developed such that it can be easily used by disabled and elderly people. Thus, it is important to develop software taking into consideration "accessibility" in order for disabled and elderly people to be able to use software.

For web sites (user interfaces implemented on the basis of HTML in many cases), there are many guidelines and evaluation tools for accessibility [1][2], and they are widely used. Guidelines represent checklists for developing accessible web sites, and evaluation tools automatically evaluate the accessibility of the target web sites. On the other side, for application software (user interfaces implemented on the basis of programming languages except for HTML), there are some guidelines [3][4], but, evaluation tools are very few.

In our research, to facilitate the development of accessible software, we propose a method of evaluating the accessibility of GUI software. In our method, source programs of the target software are analyzed and compared with guidelines, and problems in terms of accessibility and indications on how to modify them are shown. Here, we focus on the Java programming language and the Swing package in Java.

2. Accessibility

Accessibility means that various people can use information technology (IT) and services. That is, disabled and elderly people, as well as unimpaired people, can use them without difficulty.

For example, it is often difficult for people with weak eyesight and color-impaired people to accurately recognize small characters and colors on a display. Also, it is impossible for blind people to see the characters and images on a display. For such people, several support tools are developed. Thus, to improve the accessibility of software, various kinds of software help disabled people.

Examples of support tools for disabled people are screen readers and braille displays. Screen readers are software for reading text on a display. Braille displays are devices for representing the text on the display in braille. Using these tools, blind people can read text on the display.

In addition to making these tools more available, it is important to configure the color, the contrast and the size of characters appropriately and make all contents operable using the keyboard.

2.1. Guidelines

There are some guidelines for improving accessibility. Section 508 of the Rehabilitation Act [5] and W3C Web Contents Accessibility Guidelines (WCAG) [2] are well known.

The Rehabilitation Act was established by the government of the United States, and Section 508 is included in it. Section 508 stipulates that electronic and information technology that the government develops, purchases or uses must be accessible to disabled people. Software applications, operating systems, web-based intranet and internet information and applications, and various devices that the government purchases must satisfy the guideline "Electronic and Information Technology Accessibility Standards". In this standards, functions must be included in products that are to be accessible to disabled people are concretely described. As a result this law, the accessibility of various products rapidly improved.

WCAG was established by the World Wide Web Consortium (W3C) and is a guideline for making web contents accessible to disabled people. This is the most well-known accessibility guideline for web content in the world, and the guidelines of various companies and associations are developed on the basis of WCAG.

Thus, recently, the importance of accessibility has become widely acknowledged, and it is essential to develop software that is accessible.

2.2. Java Accessibility API

For Java programs, Java Accessibility API [6] is available in Java Swing.

Java Accessibility API defines connections between components of GUIs and assistive technologies that access these components. Using this API, disabled people can access software with Java Swing using support tools such as screen readers. Thus, to improve the accessibility of Java software, it is important to use this API.

In connection with Java Accessibility API, Java Access Bridge [7] is available. This is a tool for applying the assistive technologies of Java Accessibility API. That is, using Java Access Bridge, Java Accessibility API is can be used in various support tools. Java Access Bridge enables support tools to interact with Java Accessibility API.

3. Typical accessibility guidelines in our method

In our method, the accessibility of software is evaluated on the basis of the accessibility guidelines. The accessibility guidelines used in our method are developed by integrating guidelines of various companies and associations, such as Developing Accessible JFC Applications [3] and IBM Java accessibility checklist [4]. The typical guidelines of accessibility in our method as follows:

Add descriptions to widgets
If screen readers and braille displays can be made available for software, blind people can use such software. To realize this, it is necessary to add names and text descriptions to widgets.

Make all widgets operable with keyboard
It is difficult for blind and physically disabled people to operate widgets by the mouse. If all widgets on GUI are made operable using the keyboard, these people will be able to use the software. To realize this, it is necessary to use methods for adding mnemonics to widgets and to use classes for defining the order of focus to widgets using the Tab key.

Make colors and contrasts recognizable
It is difficult for color-impaired people to recognize a certain color, and for people with weak eyesight to read the text when the contrast between the background and foreground colors is weak. If such users can select the colors used and adjust the contrast by themselves, they will be able to recognize the contents of the software. To realize this, it is necessary not to use the methods for setting background and foreground colors.

Make size of text adjustable
It is difficult for people with weak eyesight to recognize small text. Also, unimpaired people may often not recognize small text correctly. If users can adjust the text size by themselves, they will be able to read text easily. To realize this, it is necessary not to use method for setting font of text.

Describe all content with text
Screen readers and braille displays cannot represent images. Therefore, blind people cannot recognize content represented only by images. Consequently, text descriptions must be prepared for each widget, so that blind users will be able to recognize the content using support tools.

Use Java Swing
Java Accessibility API is available for Java Swing package. To develop accessible software, GUIs are developed using not widgets without Accessibility APIs like Java AWT package but widgets with Accessibility APIs like Java Swing package.

All widgets need not satisfy all elements of the guidelines. For example, button and text field widgets are required to satisfy the element "Make all widgets operable with keyboard", because they are operated by the user. On the other hand, label widgets need not satisfy this element, because label widgets are used only for displaying text and images, and do not need to be operated by the user. Thus, whether each element of the guidelines is required to be satisfied or not depends on the type of widget. Also, there are concrete methods and classes that must be evaluated for each element of the guidelines. Table 1 shows examples of the types of widgets that must satisfy each element and examples of concrete methods and classes for evaluation in source programs for each element.

Table 1. Examples of evaluated methods and classes

Elements of guideline	Target types of widgets	Targets of evaluation
Add descriptions to widgets	JButton, JComboBox JTextField	"setAccessibleName()" method (use) "setAccessibleDescription()" method (use)
Make all widgets operable with keyboard	JButton, JComboBox JTextField	"setMnemonic()" method (use) "FocusTraversalPolicy" class (use)
Make colors and contrasts recognizable	All types of widgets	"setForeground()" method (not use) "setBackground()" method (not use)
Make size of text adjustable	JButton, JLabel, JTextField	"setFont()" method (not use)
Describe all content with text	JButton, JLabel	"setIcon()" method (not use)

4. System architecture

Figure 1 shows the architecture of our system named EVA (Evaluation of Accessibility). The evaluation of the accessibility in EVA consists of three steps.

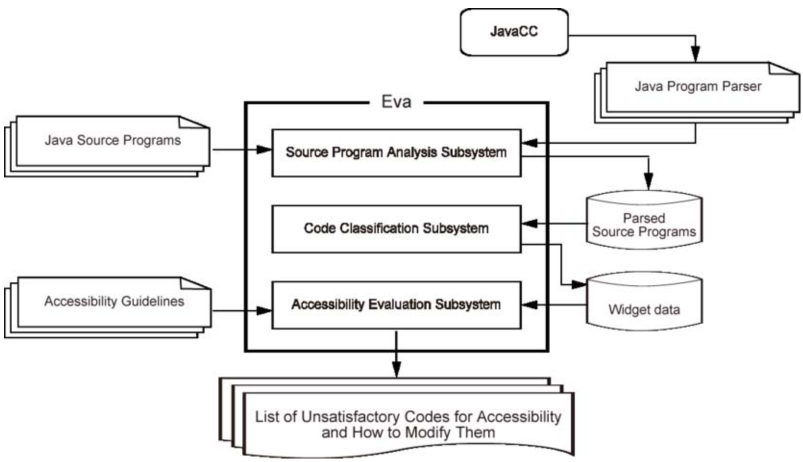


Figure 1. System architecture

4.1. Analysis of source programs

First, EVA analyzes the source programs of the target software. In this analysis, JavaCC [8] is used. JavaCC is a compiler compiler for Java programs. Using JavaCC, a parser of Java programs is generated, and source programs of the target software are analyzed using the generated parser. In this step, source programs are converted to an appropriate format for extracting the widget data of GUIs.

4.2. Classification of widget data

In this step, EVA reads the parsed source programs and extracts the widget data of GUIs. Extracted widget data are as follows: types of widgets, variable names of widgets, meth-

ods invoked from each widget, methods using widgets as parameters, and classes including these methods.

Also, to add extra methods and attributes to widgets, developers often create original widgets by inheriting widgets in the Java Swing package. Thus, EVA analyzes the relationships between classes in terms of inheritances. Then, EVA generates an Inheritance Relationship Graph (IRG). An IRG represents the relationships between parent classes and child classes. Figure 2 shows examples of a part of IRG.

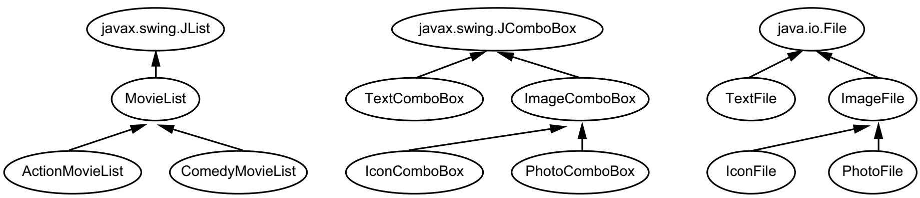


Figure 2. Examples of IRG

In an IRG, nodes represents classes, and arcs represents inheritances from child classes to parent classes. For the example in Figure 2, "javax.swing.JList" class is the parent class of "MovieList" class.

In Java programs, classes can be classified into those defined in the programs and those defined in Java. Also, all classes inherit "java.lang.Object", that is, all classes inherit some kind of class defined in Java. Thus, EVA traces an IRG from leaf nodes to their parent nodes and finds the first node that represents a class defined in Java. If the found class is a class of widgets in Java Swing or Java AWT packages, EVA evaluates classes inheriting the found class as widgets. For the example in Figure 2, classes "ActionMovieList", "IconComboBox" and "IconFile" are defined in the programs, and classes "javax.swing.JList", "javax.swing.JComboBox" and "java.io.File" are defined in Java. That is, the "javax.swing.JList" class is the first node of classes defined in Java traced from the "ActionMovieList" class, the "javax.swing.JComboBox" class is the first node of classes defined in Java traced from the "IconComboBox" class, and the "java.io.File" class is the first node of classes defined in Java traced from the "IconFile" class. Also, the "javax.swing.JList" class and "javax.swing.JComboBox" class are classes of widgets in Java Swing, whereas the "java.io.File" class is not a class of widgets in Java Swing. Therefore, EVA evaluates objects of the "ActionMovieList" class as widgets belonging to the "javax.swing.JList" class and objects of the "IconComboBox" class as widgets belonging to the "javax.swing.JComboBox" class. However, EVA does not evaluate objects of the "IconFile" class as widgets.

4.3. Evaluation of accessibility

4.3.1. Evaluation on the basis of guidelines

In this step, first, EVA compares the extracted widget data of GUIs with the guidelines. EVA compares each element in the guidelines with each type of widget extracted as described in 4.2. When a widget is required to satisfy an element of the guidelines, EVA confirms the used methods of the widget and the classes required to be used for the widget and evaluates whether the widget satisfies the element, that is, whether the methods

and classes should be used or not and whether the required methods and classes are being used or not.

4.3.2. Show of results of evaluation

After evaluating the accessibility of software as described in 4.3.1, EVA shows the results. These results are shown for every element that exhibits problems classified into either of the following two types of indications; requirement of satisfaction and requirement of attention.

Requirement of satisfaction
This type of indication is shown, when there are problems that developers are required to modify. For example, methods "setAccessibleName()" and "setAccessibleDescription()" in Table 1 are required to be used for widgets. When there are widgets for which these methods are not used, this indication is shown, and the use of these methods is indicated. In addition of the use of methods, when there are widgets to which mnemonics have not been added as necessary, and when widgets that do not have accessibility APIs like Java AWT are used, this type of indication is shown.

Requirement of attention
This type of indication is shown, not necessarily for always problems, but whenever a confirmation is necessary. For example, methods "setBackground()" and "setForeground()" in Table 1 should not be used. However, even if these methods are used, there are cases where no problems exist depending on the selection of color. Thus, EVA indicates that developers should reconfirm the usage of colors and contrasts. In addition to these uses of methods, when the order of focuses to widgets is defined, this type of indication is shown to confirm whether the appropriate order is defined or not.

5. Example

For the window shown in Figure 3, an example of evaluating the accessibility is described. Figure 4 shows part of the source program of the window in Figure 3.

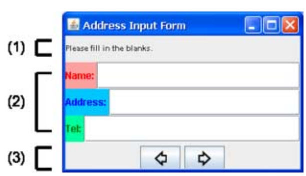


Figure 3. Example of a window before modification

(4)

```
public class AddressInputForm extends JFrame {
  public AddressInputForm() {
    topLabel = new JLabel("Please fill in the blanks.");
    ......
    nameLabel = new JLabel("Name: ");
    nameField = new JTextField();
    ......
    addressLabel = new JLabel("Address: ");
    addressField = new JTextField();
    ......
    telLabel = new JLabel("Tel: ");
    telField = new JTextField();
    ......
    setSize(300, 200);
    setVisible(true);
  }
```

Figure 4. Part of the source program of the window

There are problems in terms of accessibility in Figures 3 and 4, that is, (1)-(4) are parts with problems.

In (1) in Figure 3, very small text is used on the label widget. It may be difficult for people with weak eyesight to read the text, and unimpaired people also might not read the text correctly because of the small size and settings of the display.

In (2) in Figure 3, the background and foreground colors on the label widgets have been changed from the default color. However, the contrast between them is low, which may make it difficult for color-impaired people and people with weak eyesight to read the text.

In (3) in Figure 3, icons are used on the button widgets. However, support tools, such as screen readers and braille displays, cannot recognize the meanings of the icons, if text descriptions have not been added to them. As a result, it will be difficult for blind people to recognize the meanings of the icons. Because of the size and settings of the display, it may also be difficult for people with weak eyesight to recognize the icons.

In (4) in Figure 4, there is no use of the accessibility API for describing each widget with text. By adding names and descriptions to each widget using the accessibility API, the Java software will be able to interact with support tools smoothly. In other words, it is difficult for disabled people to use support tools without the accessibility API.

(1) Example of analysis of source programs

In evaluating accessibility, first, EVA analyzes the source programs of the target software and extracts widget data of GUIs. Some of results of the analysis of the window in Figure 3 and the extraction of widget data for GUIs are shown in Table 2.

Table 2. Some of extraced widget data for GUIs

Types of widgets	Variable names	Used methods (Line number)
JLabel	topLabel	setFont (line: 16)
JLabel	nameLabel	setForeground (line: 23), setBackground (line: 24)
JTextField	nameField	
JButton	backBut	setIcon (line: 53)
JButton	nextBut	setIcon (line: 56)

(2) Example of evaluating accessibility

EVA assesses the accessibility of each widget on the basis of accessibility guidelines, such as Table 1. In this evaluation, EVA identifies which elements of the guidelines should be satisfied on the basis of the types of each widget. Then, EVA evaluates whether appropriate methods and classes are used for each widget.

Table 3. Widgets with problems in terms of accessibility

Widgets	Unsatisfying elements of guidelines
JLabel topLabel	• Make size of text adjustable
JLabel nameLabel	• Make colors and contrasts recognizable
JTextField nameField	• Adde descriptions to widgets • Make all widgets operable with keyboard
JButton backBut, nextBut	• Describe all content with text

In the example of this evaluation, the widgets with problems listed in Table 2 were found, as shown in Table 3.

As a result of the evaluation, EVA shows the problems and indications on how to modify them. Examples of problems and indications are shown in Figure 5. Also, an example of a window modified on the basis of this result is shown in Figure 6.

JLabel topLabel
- The “setFont()” method is used. Please confirm whether users can adjust the size of font or not. (Requirement of attention)

Label nameLabel
- The “setBackground()” and “setForeground()” methods are used. Please confirm whether the use of color and contrast is appropriate. (Requirement of attention)

JTextField nameField
- The “setAccessibleName()” and “setAccessibleDescription()” methods are not used. Please add these methods to this widget to add extra descriptions. (Requirement of satisfaction)
- The “FocusTraversalPolicy” class is not used. Please confirm whether the focus moves in the appropriate order. (Requirement of attention)

JButton backBut and JButton nextBut
- The “setIcon()” methods are used. Please add the text descriptions. (Requirement of satisfaction)

Figure 5. Examples of indications on how to modify problems

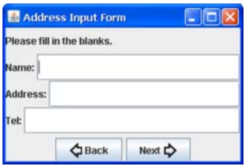


Figure 6. Example of a window after modification

6. Evaluation

To evaluate the effectiveness of our method, we confirmed the accessibility of four types of software using EVA. The four software packages were a simple database system, a music management system, a video rental management system and a graph-displaying system. The evaluation was performed in the following steps.

1. Evaluate the accessibility of the target software using EVA and Java Accessibility Helper [9]
2. Modify problems in the target software identified by EVA
3. Evaluate accessibility of the modified software using Java Accessibility Helper, and confirm remaining problems

Java Accessibility Helper is a tool for evaluating accessibility. This tool evaluates the accessibility of each widget in the target software dynamically. In this tool, problems of accessibility are classified into “P1”, “P2” and “P3”. “P1” indicates very important problems in which widgets must be modified. “P2” indicates problems where widgets should be modified. “P3” indicates elements for which widgets may be modified.

The results of this evaluation are shown in Table 4. In this table, “P1 (before)” and “P3 (before) indicate the number of P1 and P3 problems, respectively, before modifying the target software, and “P1 (after)” and “P3 (after)” indicate the number of P1 and P3 problems, respectively, after modifying the target software on the basis of the evaluation using EVA. “P2” problems in this evaluation had the same meaning as “P1” and “P3”, so they were omitted.

According to Table 4, not all problems are modified. The reasons are considered to be as follows.

Table 4. Number of problems before and after modification

System	P1 (before)	P3 (before)	P1 (after)	P3 (after)
Database	26	11	12	5
Music	20	10	8	4
Video	23	15	7	1
Graph	20	14	16	13

(1) Internal containers

Many of the remaining "P1" problems after modification were for containers use internally in JFrame and JDialog widgets. Containers are mats on which to put widgets, and JFrame and JDialog widgets are for displaying windows. That is, when JFrame and JDialog widgets are used, their internal containers are used automatically. Java Accessibility Helper evaluated these internal containers, but they usually are not described in the source programs. Since EVA does not evaluate widgets that are not described in the source programs, EVA could not find these problems. However, these internal containers do not cause actual trouble when disabled people use the software. Thus, these problems were shown as "P1", but are considered to be of low importance.

(2) Widgets defined as arrays

Widgets were often declared as variables of arrays and were generated in loop statements. In these cases, EVA could not evaluate the accessibility of those widgets. To be able to evaluate these widgets, it is necessary to improve the algorithms of EVA for analyzing the source programs.

(3) Object generation without variable declaration

Widgets were sometimes generated without declaring their variables and were put on windows. In particular, JLabel widgets were often used as this manner. JLabel widgets are widgets for displaying text or images, so it is often unnecessary to configure settings, except for text or images on them. EVA evaluates accessibility by tracing the names of the variables of widgets, so widgets without declared variables could not be evaluated. To be able to evaluate these widgets, it is necessary to improve the algorithms to indicate the declaration of variables.

However, according to Table 4, 79% of the problems at maximum could be modified on the basis of the indications presented by EVA. Therefore, our method for evaluating accessibility can be considered to be effective.

7. Related works

Several methods and tools have been proposed for the evaluation of accessibility.

As mentioned in Section 6, Java Accessibility Helper [9] is a tool for evaluating the accessibility of Java programs. In this tool, users select the target software for evaluation, and the selected software is launched. When users use the software, windows are displayed, and then, the accessibility of widgets on the displayed window is evaluated. Also, Feigenbaum and Squillace also proposed a method of evaluating the accessibility

of Java programs [10]. In their method, Java rich-client GUI-based applications are evaluated. A tool realizing this method was developed as a set of Eclipse plug-ins. This tool evaluates the accessibility of Java programs dynamically, similar to Java Accessibility Helper. However, in these methods, it is difficult to evaluate the accessibility of windows that are not displayed and to indicate concrete variable names of widgets with problems.

Freire and Fortes proposed a method of evaluating the accessibility of dynamically generated web pages [1]. In their method, the accessibility of web pages generated using XML and XSLT are evaluated by analyzing XSLT and using DTD or XSD to identify the possible XML documents that can be generated. Takata et al. also proposed a method of evaluating web pages [11]. In their method, a language for specifying accessibility guidelines and a tool for evaluation were proposed. Using their tool, it is possible to evaluate not only the accessibility but also complex syntax. However, these methods are for evaluating accessibility of web pages, not GUI software.

8. Conclusion

In this paper, we proposed a method of evaluating the accessibility of GUI software. In our method, accessibility of software is evaluated along with the accessibility guidelines by analyzing the source programs. Then, the list of unsatisfactory code for accessibility and the indications on how to modify them are shown. To assess the effectiveness of our method, we evaluated accessibility of some software. According to this assessment, many problems could be modified on the basis of the indications shown by our method, so we could confirm that our method for evaluating accessibility was effective.

Future work includes the following:

- evaluation of accessibility including structures of windows,
- evaluation of developers' own elements of accessibility,
- improvement of the algorithms for analyzing source programs to evaluate various ways of describing them.

References

[1] A. P. Freire and R. P. M. Fortes, "Automatic Accessibility Evaluation of Dynamic Web Pages Generated Through XSLT", Proc. of International Cross-Disciplinary Workshop on Web Accessibility at the Fourteenth International World Wide Web Conference (W4A at WWW2005) (2005)
[2] W3C Web Content Accessibility Guidelines: http://www.w3.org/TR/WAI-WEBCONTENT
[3] Sun Microsystems Accessibility Program: Developing Accessible JFC Applications
[4] IBM Java accessibility checklist: http://www-03.ibm.com/able/guidelines/java/accessjava.html
[5] Section 508: The Road to Accessibility: http://www.section508.gov/
[6] Java Accessibility: The Java Accessibility API: http://java.sun.com/javase/technologies/accessibility/docs/jaccess-1.3/doc/core-api.html
[7] Java Access Bridge For Windows Operating System: http://java.sun.com/products/accessbridge/
[8] JavaCC: https://javacc.dev.java.net/
[9] Java Accessibility Helper: http://java.sun.com /developer/earlyAccess/jaccesshelper
[10] B. Feigenbaum and M. Squillace, "Accessibility Validation with RAVEN", Proc. of the 2006 International Workshop on Software Quality (WaSQ'06) (2006)
[11] Y. Takata, T. Nakamura and H. Seki, "Accessibility Verification of WWW Documents by an Automatic Guideline Verification Tool", Proc. of the 37th Hawaii International Conference on System Sciences (HICSS'04) (2004)

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-145

Workflow Diagrams Based on Evidence Life Cycles

Osamu TAKAKI [a,1], Takahiro SEINO [a], Izumi TAKEUTI [a], Noriaki IZUMI [a], Koichi TAKAHASHI [a]

[a] *National Institute of Advanced Industrial Science and Technology, Japan*

Abstract. In this paper, we introduce a new language AWL of workflow diagrams for requirement analysis in large scale information system developments. AWL has a merit that one can describe concrete and accurate life cycles of evidences (evidence documents) in workflow diagrams in the language. We also introduce a tool AWDE that supports users to compose consistent workflow diagrams in AWL, by visualizing life cycles of evidences in the workflow diagrams and verifying consistency of the life cycles. As a validation of AWL and AWDE, we report an experiential result in requirement analysis in real large scale information system developments with and without AWDE.

Keywords. workflow, verification, evidence life cycle, requirement analysis

Introduction

There are several types of languages to describe workflow in requirement analysis, according to granularity of workflow that designers describe. For example, one can use activity (or statechart) diagrams in UML to describe workflow [5], [6]. However, UML activity diagrams are not suitable to describe workflow in low granularity. Moreover, these diagrams are not easy to understand for customers, because they are not familiar with system developments. There is a workflow language of describing workflow in low granularity, which is developed by Ministry of Internal Affairs and Communications of Japan [9]. However, this language is not formal, so that one can compose ambiguous workflow diagrams in the language. BPMN [3] and XPDL [14] are standard workflow languages that are defined to be easy for customers to understand workflow in low granularity. However, even they are not easy for customers to describe workflow in low granularity precisely. In order to obtain workflow diagrams in which one can describe workflow in low granularity precisely, we have developed a language of workflow diagrams based on the notion of “evidences”, which are explained in the following paragraph.

In large organizations such as large enterprises or governments, documents such as order forms, estimate sheets, specification descriptions, invoices and re-

[1] Corresponding Author: National Institute of Advanced Industrial Science and Technology, Japan; E-mail: o-takaki@aist.go.jp.

ceipts play significant roles for the purpose of feasibility, accountability, traceability or transparency of business. In such organizations, a task involves workers of different roles, and is carried out in circulating documents. Such documents are considered as a kind of evidences of the purpose above. In this paper, we call such documents "evidences".

Some evidences are triggers that request office workers to perform some tasks and others are manuals that teach workers how to perform tasks. Some workers check evidences and sign in the evidences when they accept the contents of the evidences. Therefore, today a lot of actual operations are based on evidences even if they are carried out with information systems. So, for requirement analysis in developments of large scale information systems, it is important to consider workflow diagrams in which one can describe life cycles of evidences in a concrete and precise way.

In this paper, we introduce a new language of workflow diagrams called "AWL" (AIST Workflow Language). In workflow diagrams of AWL as well as major workflow languages, control flows in workflow are described by arrows. Moreover, one can describe how to take over evidences by the arrows in workflow diagrams of AWL. A control flow of workflow and a flow of an evidence are described by an arrow. Since most operations are carried out with some specific evidences, it is rational that a single arrow denotes both of a control flow and a flow of an evidence. By describing flows of evidences with appropriate control flows, one can easily describe and understand life cycles of the evidences. We also develop a design support tool "AWDE" (AIST Workflow Development Environment), which supports to compose workflow diagrams in AWL and verifies life cycles of all evidences in the workflow diagrams automatically. This tool enable users to compose consistent workflow diagrams, by visualizing life cycles of evidences in the workflow diagrams and verifying consistency of the life cycles. By verifying consistency of the life cycles in a workflow diagram, one can often find inconsistencies of structure of the workflow diagram.

The remainder of this paper is organized as follows. In Section 1, AWL is introduced. In Section 2, we explain consistency of evidence life cycles in workflow diagrams. In Section 3, we explain AWDE, which supports one to compose workflow diagrams and verifies evidence life cycles in them. In Section 4, we report an experiential result of requirement analysis in real large scale information system developments with and without AWDE in order to show the tool enables users to take the initiative in requirement analysis in large scale information system developments, even if they are not familiar with system developments. Sections 3 and 4 will validate our methodology. In Section 5, we explain related works to our language and tool. Finally, we conclude the paper in Section 6.

1. AIST Workflow Language

In this section, we explain a language of workflow diagrams, which is called "AWL" (AIST Workflow Language). AWL is defined to be appropriate to compose workflow diagrams for human workflow easily, and to verify life cycles of evidences of workflow diagrams.

1.1. Overview of AWL

Comparing standard workflow languages, the main feature of AWL is that in a workflow diagram of AWL one can assign to each activity node a list of evidences (evidence documents) which are used in the activity. In the perspective of control-flow, workflow diagrams of AWL is similar to those in BPMN [3] or XPDL [14], which are the most standard workflow languages, or workflows in previous researches such as [10], [1] or [8]. That is, one can regard (control-flow parts of) workflow diagrams of AWL as a directed graph, whose nodes are of the following types: start, end, activity, XOR-split, XOR-join, AND-split and AND-join. For more details on control-flow of workflow diagrams, see [10], [1] or [8].

Instead of explaining AWL in detail, we explain workflow diagrams of AWL by using an example. The figure 1 is a workflow diagram describing a work of planning of a research. In the diagram, rectangles denote operations needed for planning of a research, and yellow figures denote evidences (evidence documents) used on the operations.

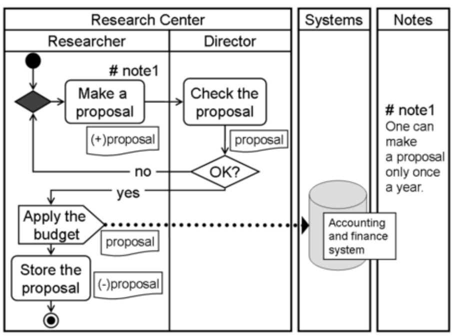


Figure 1. Example of a workflow diagram

In this example, first a researcher composes a proposal of a research, and then a director checks the proposal. If the proposal passes the checking, then the proposal is returned to the researcher and he/she applies the budget on an accounting and finance system based on the proposal, and finally the proposal is stored by the researcher. If the proposal does not pass the checking, then the proposal is returned to the researcher and he/she remakes the proposal.

1.2. Evidences

By "an evidence" in workflows one means a paper document or a data (a data file) of a document. In this paper, we regard an evidence as a paper document, which is composed, referred, re-written, judged, stored or dumped in some activities. Unlike data files, an evidence does not increase. Though one can make a copy of it, the copy is regarded not to be the same thing as the original evidence. Moreover, unlike data in a system multiple people can access simultaneously, an evidence can not be used by multiple people at the same time.

In formulating workflow diagrams, especially, those for human workflows, evidences are still very important even through a lot of paper documents are

replaced by data (data files) in information systems. In a workflow diagram of AWL, evidences used in an activity is explicitly described in the activity node, in order to describe and verify life cycles of evidences more correctly.

In the technical perspective, a list of evidences with length at least 0 is assigned to an activity node, and an evidence E is defined to be a triple $(e, \mathit{created}, \mathit{removed})$, where e is a label, and *created* and *removed* are boolean values. For simplicity, each evidence $E = (e, \mathit{created}, \mathit{removed})$ is abbreviated, as follows (see the figure 1).

(i) If $\mathit{created}$ = false and $\mathit{removed}$ = false, then we abbreviate E as "e".
(ii) If $\mathit{created}$ = false and $\mathit{removed}$ = true, then we abbreviate E as "$(-)e$".
(iii) If $\mathit{created}$ = true and $\mathit{removed}$ = false, then we abbreviate E as "$(+)e$".
(iv) If $\mathit{created}$ = true and $\mathit{removed}$ = true, then we abbreviate E as "$(+)(-)e$".

For a workflow diagram, every evidence E is described near just one activity node A, and it means that E is an evidence document used at the operation denoted by A. Moreover, if E has the $(+)$-mark, then it means that E occurs in A for the first time in the workflow diagram. On the other hand, if E has the $(-)$-mark, then it means that E is removed (stored or dumped) in A.

2. Consistency of Evidence Life Cycles in a Workflow

In this section, we explain a notion of consistency of evidence life cycles in a workflow. We also show some premises to justify our definition of the consistency, based on our observation of workflows which have been composed in real developments of information systems. Consistency property of life cycles of evidences in an acyclic workflow has been defined in [11] and [12]. We also explain a notion of consistency in a cyclic workflow.

2.1. Premises for consistency of life cycles of evidences

The phrase "consistency of a life cycle of an evidence E" usually means that there is no inconsistency in change in state of E during the time between E is created and removed. However, it is not quite easy to define consistency of life cycles of evidences from the point of view of the whole workflow. Therefore, we explain several premises for considering the consistency.

By virtue of investigation of about 400 workflow diagrams, which have been composed in developments of real large scale information systems, we have the following facts about real workflows.

- Workflows contain no loop which is considered in usual programs and assured its termination property. Moreover, as far as we know, every loop in a workflow diagram contains an arrow which should be called a "pass-back".
- Every information described in a workflow diagram, except which in a note, is treated as a "static" information. In particular, no information about operations after a job turns back via a pass-back arrow.

A pass-back arrow is an arrow making a job turn back to a node which the job has been before.

For example, in the figure 1, suppose that the proposal does not pass the checking by the director. Then, we can consider the job with the proposal is

turned back to the activity node where the researcher should make a proposal via the pass-back arrow labeled "no". In this case, the researcher should revise or dump the proposal, and hence, the evidence "(+)proposal" is not appropriate. However, there is actually no modification for such a case in a workflow diagram.

It maybe possible to reconstruct such a workflow so that there is not such an inconsistency above which occurs on the target of the pass-back arrow. Thus, one can select a policy which prohibit inconsistencies on the targets of pass-back arrows like the example above. However, in many cases, this constraint is so strong that workflow diagrams have too large size or too complex structure. Therefore, we do not select such a policy, that is, we consider the following premise.

Premise 1: We do not consider inconsistency between the target and the source of a pass-back arrow in a workflow.

Next, we consider several cases where defects of descriptions of evidences come from inconsistencies of structures of workflows. So, we also give the following premise.

Premise 2: It makes no sense to define consistency of life cycles of evidences in a workflow which has inconsistencies in its structure.

We will explain consistency of the structure of a workflow in Section 2.2.

Finally, we consider that an evidence is a paper document, and give the third premise, as follows.

Premise 3: No evidence increases or decreases.

For example, when one makes copies of an evidence E, we do not consider that E increases, but consider that new evidences $E'_1, \dots, E'_n$ are created and that they are different from each other.

On the basis of the premises above, we will define consistency of life cycles of evidences in a workflow. In order to do so, we explain correctness of acyclic workflows, which is a consistency property of structure of acyclic workflows.

2.2. Correctness of Workflows

An inconsistency of structures of workflows comes from a wrong combination of XOR-split/join nodes and AND-split/join nodes. Such inconsistencies are known as "deadlock" and "lack of synchronization" [10]. An acyclic workflow which is deadlock free and lack of synchronization free is said to be "correct" [1].

Definition 2.1 For an acyclic workflow W, an instance of W denotes a subgraph V of W that satisfies the following properties.

1. V contains just one start node. Moreover, for each node x in V, there exists a path on V from the start node to x.
2. If V contains an XOR-split c, then V contains just one outgoing-arc of c.
3. If V contains a node x other than XOR-split, then V contains all outgoing-arcs of x.

Definition 2.2 Let W be an acyclic workflow.

1. An instance V of W is said to be deadlock free if, for every AND-join r in V, V contains all incoming-arcs of r.

2. An instance V of W is said to be lack of synchronization free if, for every XOR-join m in V, V contains just one incoming-arc of m.

Definition 2.3 An acyclic workflow W is said to be correct if every instance V of W is deadlock free and lack of synchronization free.

The correctness property above not only assures consistency of a given workflow in the viewpoint of control-flow, but also enables one to define consistency of life cycles of evidences, by using all instances in the workflow diagram.

2.3. Definition of consistency of evidence life cycles

We now define consistency of life cycles of evidences in a workflow. We first define the consistency property for acyclic workflows and then define that for general workflows.

Definition 2.4 For an acyclic workflow W, W is said to be "consistent for life cycles of evidences" if W is correct and, for every instance V of W, for every activity node A in V and for every evidence E on A, there exists just one sequence L of arrows in V $L := (A_1 \xrightarrow{f_1} A_2 \xrightarrow{f_2} \cdots \xrightarrow{f_{n-1}} A_n)$ such that
(i) every activity node in L contains E,
(ii) only A_1 has the $(+)$-mark of E,
(iii) only A_n has the $(-)$-mark of E and
(iv) some A_i is A.

We have shown that, for a correct workflow W, W is consistent for life cycles of evidences, if and only if, every evidence in W is first created explicitly, and it is kept with no increase and no decrease, and it is eventually removed explicitly in every case (see Lemma 3.6 in [11]).

Here we define consistency of evidence life cycles in a general workflow diagram which may have some loops. Note that in Section 2.1 we gave a premise not to consider inconsistencies between the target and the source of a pass-back arrow.

Definition 2.5 A workflow W is said to be "consistent for life cycles of evidences" if the acyclic workflow obtained from W by deleting all pass-back arrows in W is consistent for life cycles of evidences.

3. AIST Workflow Development Environment

In this section, we explain a tool AWDE (AIST Workflow Development Environment) in short. The following figure 2 describes a snapshot of composing a workflow by AWDE and its architecture. AWDE has a plug-in tool called "AWV" (AIST Workflow Verifier), which is a verification program of workflow diagrams. We have developed AWDE based on Microsoft® Visio®. Users can compose workflow diagrams of AWL in the same way as Microsoft® Visio®. They can also verify workflow diagrams which they have composed or they are composing at any time by only one-click.

We now explain how AWDE verify workflows briefly, as follows.

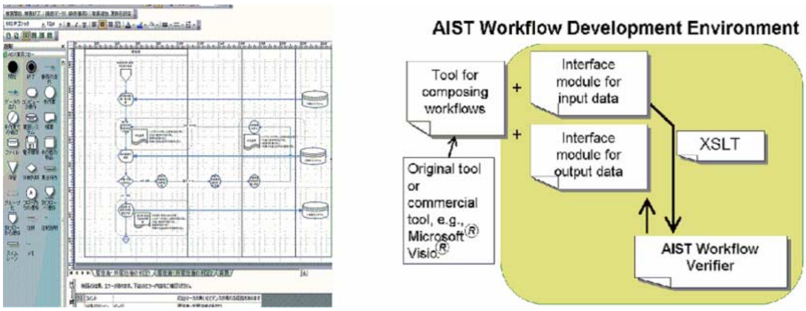


Figure 2. Snapshot and Architecture of AWDE

1. When a user clicks the button, the interface module verifies syntactical correctness of workflow diagrams. If the module detects syntactical errors of them, AWDE displays the errors immediately.
2. If the workflow diagrams have no syntactical error, then the XSLT in the figure 2 converts XML expressions of workflow diagrams generated by Visio® into our own XML expressions of those.
3. AWV receive the XML expressions $\mathbf{W}$ of workflow diagrams as input data, and it verifies several graph-theoretical properties of workflow diagrams (e.g. connectivities of them). If there is a workflow in $\mathbf{W}$ which does not satisfy the graph-theoretic properties, then AWV returns error messages.
4. If every workflow in $\mathbf{W}$ satisfies the graph-theoretic properties, then AWV detects all pass-back arrows in $\mathbf{W}$ and translates $\mathbf{W}$ to appropriate acyclic workflows $\mathbf{W}^*$.
5. AWV extracts all instances from each workflow in $\mathbf{W}^*$ as well as verifies correctness of it. If there is a workflow in $\mathbf{W}^*$ which is not correct, then AWV returns error messages.
6. If every workflow in $\mathbf{W}^*$ satisfies correctness, then AWV verifies consistency of life cycles of evidences in $\mathbf{W}^*$. If there exists a workflow which is not consistent for life cycles of its evidences, AWV returns the list of counterexamples of the consistency.
7. When the interface module for output data receives an XML file of error messages or counterexamples above, AWDE displays the results.

For more details on the way to abstract instances, that to verify correctness or that to verify consistency for life cycles of evidences, see [11] or [12]. Here we only explain one theorem assuring of ability of AWDE.

Theorem ([11], [12])
(1) For every workflow diagram W, AWDE can determine whether or not the acyclic workflow diagram W^* translated from W is correct.
(2) If W^* is correct, AWDE detects counterexamples of consistency of evidence life cycles, unless W is consistent for evidence life cycles. In particular, W is consistent for life cycles of evidences, if and only if, AWDE returns the empty list.

4. Experience

In order to validate our verification methodology of workflow diagrams based on evidence life cycles, we report an experiential result, which we obtained through requirement analysis for real large scale information systems in 2007 that employed AWDE and another one that did not employ AWDE.

4.1. Two system developments

We explain two information systems for the experience, as follows.

N: Accounting and finance system for a local government.

Development scale: About 1000,000,000 yen. [2]
Number of files of workflow diagrams: 120.

T: Accounting and finance system for an independent administrative institution.

Development scale: About 850,000,000 yen.
Number of files of workflow diagrams: 187.

AWDE was employed in the system development of N, while it was not employed in that of T.

Both of N and T are large scale information systems. We can not compare the precise scales of them, but, one would be able to consider that they have similar scales in the viewpoint of the costs of their system developments.

4.2. Comparison result of personal costs

As the main comparison criterion, we deal with personal costs of composing workflow diagrams in N and T, respectively. Especially, we are concerned with kinds of people who composed the workflow diagrams and length of sessions of composing them. We show personal costs of composing workflow diagrams, as follows.

Personal cost of composing workflow diagrams in development of N

Two office workers (Total of 12 man-month)

- Five months for composing (As Is) workflow diagrams.
- One month for reviewing workflow diagrams above. [3]

Two system engineers (Total of 2 man-month)

- One month for reviewing workflow diagrams above.

Personal cost of composing workflow diagrams in development of T

Four system engineers (Total of 12 man-month)

- Three months for composing and reviewing (To Be) workflow diagrams.

[2] One yen corresponds to about 1/160 euro around that time.

[3] In both of N and T, several users, who were office workers and did not directly get involved composition of workflow diagrams, were interviewed for reviewing workflow diagrams.

Two system administrators (Total of 6 man-month)

- Three months for composing and reviewing workflow diagrams above.

The most interesting point is that designers of workflow diagrams on requirement analysis in development of N are office workers of the local government who actually work with the information system. Since they did not have any knowledge of composing workflow diagrams, one of us taught them how to compose workflow diagrams with AWDE. Before composing the As-Is workflow diagrams above, they practiced composing workflow diagrams with AWDE for about one month, while carrying out their primary tasks. In the beginning, they often composed workflow diagrams with many defects. However, AWDE enabled them to modify such defects, and they became progressively to compose workflow diagrams with few defects. Moreover, since they could clearly recognize flows of evidences in workflow diagrams of AWL, they could often notice defects in life cycles of evidences before checking them by the tool. On the other hand, on requirement analysis in development of T, professional system engineers and administrators composed workflow diagrams.

Although it is difficult to compare the personal costs of composing workflow diagrams, at least, we could confirm that office workers, who composed workflow diagrams with AWDE in development of N, could ensure about the same level of describing workflows on workflow diagrams as that by professional system engineers and administrators in development of T. This means that AWDE is certain helpful for enabling office workers, who know the whole works using information systems well, but are not familiar with system developments, to analyze jobs with information systems as professional system engineers.

5. Related Works

There have been developed a lot of workflow languages such as BPMN [3] and XPDL [14]. Our language AWL is developed in order to describe human workflows and the main feature of AWL is that one can easily describe and understand life cycles of evidences in workflow diagrams (see Section 1). Our past papers [11] and [12] also discuss workflow diagrams. Those papers use a terminology 'flow' which denotes 'arrow' in this paper.

There have been several investigations of document-centric workflows such as [4] [2], [7] and [13]. However, our definition of consistency of evidence life cycles in a workflow diagram, which is based on instances of the workflow diagram, is very clear, and AWDE can verify consistency of structure and evidence life cycles over a workflow diagram rigorously at least as far as the workflow diagram is acyclic. Moreover, our language is easy for workflow designers to describe what evidences are used in an activity described in a workflow.

Compared with other verification tools, AWDE can verify new properties of workflow diagrams, that is, it verifies correctness and consistency of evidence life cycles in a workflow diagram.

6. Conclusion and Future Work

In this paper, we have introduced a language AWL of workflow diagrams and consistency of evidence life cycles in workflow diagrams, and a design support tool AWDE. AWL has a merit that one can describe concrete and accurate life cycles of evidences (evidence documents) in its workflow diagrams, and AWDE can verify consistency of life cycles of evidences in workflow diagrams. We have also reported an experiential result of requirement analysis in real large scale information system developments with and without AWDE, and shown that the tool actually enabled users who were not familiar with system developments to take the initiative in requirement analysis in large scale information system developments.

Our future work is to develop new design support tools by which users can formulate other specifications such as use case scenarios and/or data models.

References

[1] W. M. P. van der Aalst, A. Hirnschall and H. M. W. Verbeek: An Alternative Way to Analyze Workflow Graphs, *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE), LNCS* **2348**, Springer (2002), 535–552.

[2] R. A. Botha and J. H. P. Eloff: Access Control in Document-centric Workflow Systems An Agent-based Approach, *Computers and Security*, **20**(6), (2001), 525–532.

[3] Business Process Management Initiative (BPMI): *Business Process Modeling Notation (BPMN) Version 1.0*, (2004).

[4] P. Dourish, K. K Edwards, A. Lamarca, J. Lamping, K. Petersen, M. Salisbury, D. B Terry and J. Thornton, Extending document management systems with user-specific active properties, *ACM Transactions on Information Systems*, **18**(2), (2000), 140–170.

[5] M. Dumas and A. H. M. ter Hofstede: UML Activity Diagrams as a Workflow Specification Language, *UML 2001, LNCS* **2185**, Springer, (2001), 76–90.

[6] H. E. Eriksson and N. Penker: *Business Modeling With UML*, John Wiley & Sons, Inc, (2000).

[7] R. Krishnan, L. Munaga and K. Karlapalem: XDoC-WFMS: A Framework for Document Centric Workflow Management System, *Conceptual Modeling for New Information Systems Technologies, LNCS* **2465**, Springer, (2002), 348–362.

[8] R. Liu, A. Kumar: An Analysis and Taxonomy of Unstructured Workflows, *Proceedings of 3rd International Conference on Business Process Management (BPM), LNCS* **3649**, Springer (2005), 268–284.

[9] Ministry of Internal Affairs and Communications of Japan: *Jichitai EA, Gyomu · shisutemu Sasshinka no Tebiki*, http://www.soumu.go.jp/denshijiti/eatebiki/index.html, (in Japanese).

[10] W. Sadiq and M. E. Orlowska: Analyzing Process Models Using Graph Reduction Techniques, *Information Systems*, **25**(2), (2000), 117–134.

[11] O. Takaki, T. Seino, I. Takeuti, N. Izumi and K. Takahashi: Verification Algorithm of Evidence Life Cycles in Extended UML Activity Diagrams, *Proceedings of The 2nd International Conference on Software Engineering Advances*, IEEE Computer Society Press, (2007).

[12] O. Takaki, T. Seino, I. Takeuti, N. Izumi and K. Takahashi: Algorithms verifying phenomena independence and abstracting phenomena subgraphs of UML activity diagrams, *Software Engineering Saizensen 2007*, Kindaikagaku-sha, (2007), 153–164, (in Japanese).

[13] J. Wang and A. Kumar: A Framework for Document-Driven Workflow Systems, *Proceedings of 3rd International Conference on Business Process Management (BPM), LNCS* **3649**, Springer, (2005), 285–301.

[14] Workflow Management Coalition (WfMC): *Workflow Management Coalition Workflow Standard: Workflow Process Definition Interface - XML Process Definition Language (XPDL)*, (WfMC-TC-1025), Technical report, Workflow Management Coalition, Lighthouse Point, Florida, USA (2002).

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-155

Automatic Conversion from Specifications in Japanese into Class Diagrams in UML

Ryuichi SAITA[a], Chiaki KUBOMURA[b] , and Hiroyuki KAMEDA[c,1]
[a]*Hachioji Campus, Nippon Engineering College*
[b]*Yamano College of Aesthetics, and* [c]*Tokyo University of Technology*

Abstract. This paper describes some automatic conversion methods and a system which converts Japanese specifications to UML class diagrams. The characteristics and limitation of the methods was also discussed in terms of two experiments with use of specifications in some UML textbooks. The results of evaluation experiments showed the fundamental validity of the methods and system.

Keywords. Text to Class Diagrams Conversion, Natural Language Processing

Introduction

UML has been applied to large scale of software system designing since around 1990. But generally UML diagrams are manually described. This means UML is not always utilized effectively. From this point of view, we discuss some methods to convert Japanese document on specifications of software system into UML diagrams automatically by choosing class name candidates to create model elements in XMI format with use of some additional information, e.g. frequency, and then produce UML diagrams. Evaluation results are also reported in this paper.

1. Automatic Conversion Approach from Specifications to Class Diagrams

In general conversion from specifications to class diagrams has been done manually by humans. When a human draws class diagrams, he/she tries to consider various relationships among class candidates. Concretely, a human tries to relate the model elements such as "Class", "Relation", "Attribute", and "Operation", where "Class" is a concept in the system, "Relation" means relation between classes, "Attribute" is data that the object owns, and "Operation" is a behavior that the object owns. With the following procedure described below, we think that computers can do the intelligent procedure to convert linguistic specifications into UML class diagrams. The procedure we designed is as follows:

[STEP 1] Morphological analysis processing
[STEP 2] Class candidate extraction processing with use of Japanese grammatical information
[STEP 3] Use of both XML subset and UML modeling tool to draw class diagrams

---
[1]Hiroyuki KAMEDA: School of Computer Science, Tokyo University of Technology, 1404-1 Katakura, Hachioji, Tokyo 192-0982, JAPAN; email:kameda@cs.teu.ac.jp

2. Class Extraction Methods and a Relation Discovery Method

The first method (Method 1) we adopted uses modification relations between classes and attributes, which relations are described by a Japanese postposition word “no”. Modifying a noun in Japanese has a role of “explaining attribute of the noun”. That is, there might exist relations between classes and attributes, when both classes and attributes are described in Japanese specifications and combined with the postposition “no” in a sentence. Syntactic relation between class and attribute combined with “no” is as follows: [Class word] + [Postposition “no”] + [Attribute word]. In Japanese, a noun just before the "no" will be a class candidate.

The second method (Method 2) is to extract class with use of word frequency, which pays attention to relations between nouns and their occurring frequency. When content of a specification is too poor, it is infeasible to understand what kind of system should be made. Therefore a good amount of description for other persons to understand the content of the system is necessary to a document of specification. This means we can expect that the more important a description of a system is, the more frequently and repeatedly the description appears in the specification. Therefore important information is expected to be frequently written in a specification. Based on this idea, we set a threshold to extract nouns that exceed fixed frequency in the document as a class. As we also assume that core of important description is made of nouns in many cases in Japanese documents.

The third method (Method 3) is to extract classes with use of database of classes. This method uses commonly-observed features (expressions) among documents on the same domain. First, nouns which tend to be a class are in advance registered in a database. Then, all nouns in a specification are checked by database. If a noun is matched with one of nouns in a database, then the noun is a class in this case. This method can be used as an extracting method based on domain-specific knowledge.

After picking out classes by the above methods, what we have to do next is to discover existence of relation between the classes. In our approach, we assume that there exists a relation between the classes when more than two different classes are included just in one sentence. Based on this idea, we decided to pick out all classes included in one sentence and let them connect with each others.

3. Prototype System

We implemented a prototype system to convert a Japanese specification into class diagrams in UML automatically. The system picks out classes, discovers the relation between them and outputs these results in the XML subset format. This system adopts all the methods mentioned above in the previous sections 2 to extract classes and relations automatically. The system consists of four parts, i.e., input support module, class extraction part, model element conversion part, and morphological analysis system module. Figure 1 shows an overview of the prototype system.

4. Evaluation Experiments

To verify the validity of the methods in section 2, we did two evaluation experiments as follows. Database used for the method 3 was created by a few graduate students.

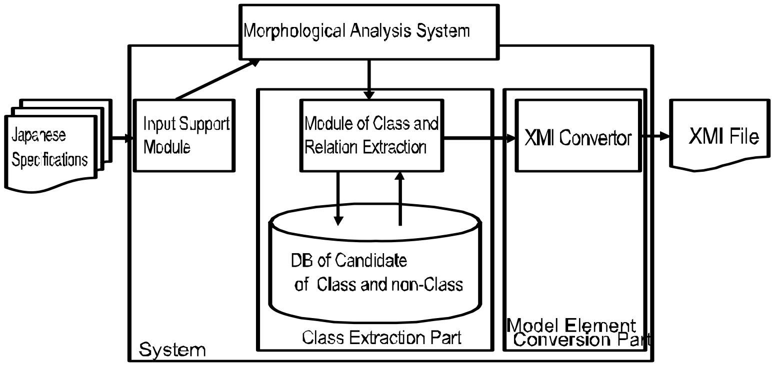


Figure 1. Overview of the prototype system

First, graduate students read specifications, event flow diagram and scenarios, and then picked up and registered all nouns which were likely to be class manually.

[Evaluation experiment 1] (Recall and Precision)

1. Goal: We confirm that classes and their relations can be extracted from Japanese specifications by our proposal methods.
2. Materials: The example of specification was in a textbook of UML [1].
3. Indices: Recall rate and precision. Recall rate f_{r1} is the ratio of the number of correct class selected by the methods to the number of all class candidates in the sentence. Precision f_{p1} is the ratio of the number of correct classes to the number of class candidates selected by the methods. Recall rate and precision are also estimated in terms of relation discovery. Recall rate f_{r2} is the ratio of the number of correct relations selected by the methods to the number of all relation candidates in the sentence. Precision f_{p2} is the ratio of the correct number of relations to the number of relation candidates by the methods.
4. Results: Table 1 is the result of recall rate f_{r1} and precision f_{p1} by the three methods described in the section 2, where the "integration method" in the table 1 is a hybrid method into which all three methods method 1, method 2 and method 3 were integrated. Table 1 shows that applying the three superficial methods separately is not so good, but the integrated method gives fairly good performance. Based on this result, we adopted only integrated method to estimate recall rate f_{r2} and precision f_{p2}, and the result is Table 2 which shows the results of both recall rate f_{r2} and precision f_{p2} are now quite lower than practical level so far.

Table 1. Results of class extraction

	Method 1	Method 2	Method 3	Integrated Method
Recall Rate f_{r1}(%)	66.7	66.7	50.0	83.3
Precision f_{p1} (%)	66.7	33.3	37.5	31.3

Note: Methods 1-3 are described in the section 2,
and integrated method is one combined with all the three methods.

Table 2. Results of relation discovery

Recall rate f_{r2} (%)	40.0
Precision f_{p2} (%)	7.0

5. Considerations: Every Method 1, 2 and 3 did show poor performance, as we expected, because they are too primitive to extract semantic information from text. Nevertheless our prototype system by the integrated method could pick out classes at more than 80% recall rate, therefore it may be feasible to use our system by the integrated method as UML class diagram support system. On the other hand, precision of the system is lower than 40%. This is because there is empirically a relation that when recall rate goes high, precision goes low, and vice versa. But for UML class diagram support system, it is important for human users to choose class candidates as many enough as possible. From this point of view, the integrated method shows quite a good performance. On the other hand, performance of relation discovery is not so good in the Table 2. The reason is that relation discovery should be done by using more syntactic and semantic information. Therefore more investigation is strongly necessary.

[Evaluation experiment 2] (making class diagram from a Japanese specification)

1. Goal: We compare the results of making class diagram from Japanese specification by our proposal system with those by human beings.
2. Materials: The example of specification was in a textbook of UML [2].
3. Indices: We measured processing time spent by the prototype system on making the class diagram from Japanese specifications, and also the time spent by human users respectively. Moreover, class diagrams made by the system and human were compared with each other. We measured time between the onset time when we started to run the prototype system to output the XMI form from Japanese specifications, and the offset time when UML modeling tool finished converting the XMI form as a class diagram. We used JUDE as UML modeling tool. On the other hand, we measured time for human users to make and draw class diagrams from a Japanese specification. Human users used manually drawing functions of JUDE to draw class diagram. Subjects of this experiment are three people (two graduate students and one of university staff). We guided subjects to make class diagram only on relations and classes. Japanese specification we used in this experiment is example 2 in the following.
4. Results: Table 3 shows consumed time how long did it take for prototype system and human users to finish making class diagram. Figures 2 and 3 below are made by our proposal system and a subject, respectively. Figure 4 is an example of correct class diagram of text in the textbook[2] (Figure 5).

Table 3. Consumed time of making class diagrams by system and human

	System	Subject 1	Subject 2	Subject 3
Consumed Time [sec]	20	296	393	319

5. Considerations: We compared the work human being did and the work prototype system did when making class diagrams. When making class diagram, human being was generally considering the background of a specification given, problems which are likely to happen, and also visibility of class diagrams. Therefore, the human being took more time to make class diagrams than the system. On the other hand, the prototype system doesn't consider any background, any problem and the visibility of diagrams, so it takes less time to make class diagrams than human. When the focus is onto the classes and their relations, human draws class diagrams of minimum set of classes and relations. On the other hand, prototype system can't rigidly chose classes and their relations.

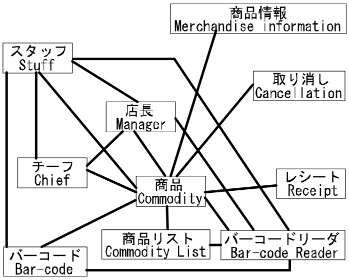


Figure 2. Class diagram by system

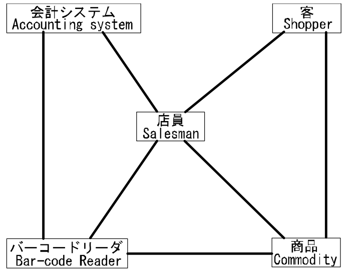


Figure 3. Class diagram by a subject

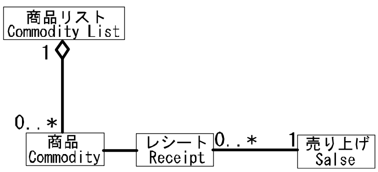


Figure 4. Answer of class diagram in textbook[2]

– 販売処理ユースケースのイベントフロー．
– 商品種類および商品が登録されていること．
– お客さんが来るとレジで，スタッフ，チーフまたは店長はバーコードリーダーで商品のバーコードを読み込みます．
– システムは，バーコードリーダーから読み取った情報をもとにまず商品リストを参照して商品の特定をします．
– システムは，商品種類リストから，商品を特定して，商品情報を取得します．
– システムは，レシートに商品名と価格を追加して，商品の価格を売り上げに追加します．
– スタッフ，チーフまたは店長は，すべての商品をバーコードリーダーで繰り返し読み込みます．
– お客さんが一部，または全部の商品の購入を取り消したときは取り消しの処理を行います．
– バーコードの読み込みができないときは，同種の商品と取り替えます．

Figure 5. Input text to our System[2]

5. Related Works

Research [3] on how to convert natural language into pattern expressions, e.g. graphical charts, is one of the oldest. There are also many studies of conversion between natural languages and UML[4-7]. For examples, a method of conversion from natural language expressions of use case description to DFD diagrams via intermediate language[4], an editor for conversion from pre-edited natural language expressions to UML[5]. Studies like ours to convert raw expressions in Japanese into UML class diagrams are few in the previous studies, because of difficulty in handling Japanese text.

6. Concluding Remarks

This paper discussed new automatic conversion methods from Japanese specifications to class diagrams, and also proposed prototype system. Moreover evaluation experiments were also reported to show fundamental validity of our proposal.

References

[1] M. SENDO: Easy UML, SHOEISYS, 2001,107.
[2] A. Takemasa: UML for beginners, Natsume, 2005.211-259
[3] Schank, R.C.: Conceptual Information Processing, Elsevier, New York, 1975.
[4] Y. MASUZAWA: On Affordance Approach for System Modeling Methodology, Information Systems Society of Japan, 2006.
[5] Y.CHUBACHI, Y.MATSUZAWA and H.OHIWA: Scenario-Based Visual Analysis for Object Modeling, The 44th Programming Symposium, Information Processing Society of Japan, 2003.
[6] F. Meziane, N. Athanasakis and Sophia Ananiadou: Generating Natural Language specifications from UML class diagrams, Springer, Vol.18, 2008,1-13.
[7] M. KIKUKAWA, J. SHIROGANE and Yoshiaki FUKAZAWA: Program Understanding by Merging Event Flow using Natural Language Processing Technology, IEICE Technical Report, KBSE2002-30, 2003, 31-36.

This page intentionally left blank

Architecture of Knowledge-Based Systems or Shells Including Agents

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-163

Autonomous Evolutionary Information Systems and Active Database Systems: A Comparative Study

Jingde CHENG [1]
Department of Information and Computer Sciences, Saitama University, Japan

Abstract. Both active database systems and autonomous evolutionary information systems are motivated by improving the passive characteristic of traditional database and/or knowledge-base systems; but they were independently developed until now. To investigate the difference between active database systems and autonomous evolutionary information systems and to find some complementary development methodologies for the both, this paper presents a comparative study of active database systems and autonomous evolutionary information systems. We comparatively discuss motivations, purposes, capabilities, logical foundations of active database systems and autonomous evolutionary information systems. We show that some facilities and their implementation techniques independently developed in active database systems and autonomous evolutionary information systems separately can be complementarily used each other and this will certainly result in progress of the both.

Keywords. Database systems, Knowledge-base systems, Event-driven rule-processing, Reasoning rule generation, Autonomous evolution

Introduction

Traditional database systems are passive, i.e., data is created, retrieved, modified, updated, and deleted only in response to operations issued by users or application programs, and the systems only can execute queries or transactions explicitly submitted by users or application programs but have no ability to do something actively and autonomously by themselves. On the other hand, traditional knowledge-base systems (including deductive databases) have the ability to deduce some implied facts from the data/knowledge by inference rules explicitly stored in the systems. However, from the viewpoint of activation and/or autonomy, traditional knowledge-base systems are intrinsically the same to traditional database systems in that data/knowledge and inference rules are created, retrieved, modified, updated, and deleted only in response to operations issued by users or application programs, and the systems only can execute tasks explicitly submitted by users or application programs but have no ability to do something actively and autonomously by themselves.

[1] Corresponding Author: Department of Information and Computer Sciences, Saitama University, Saitama, 338-8570, Japan; E-mail: cheng@ics.saitama-u.ac.jp

Until now, there are two different approaches to deal with the above problem, i.e., active database systems proposed in the area of data engineering, and autonomous evolutionary information systems proposed in the area of knowledge engineering. Both active database systems and autonomous evolutionary information systems are motivated by improving the passive characteristic of traditional database and/or knowledge-base systems. However, they were independently investigated and developed until now. To investigate the difference between active database systems and autonomous evolutionary information systems and to find some complementary development methodologies for the both, this paper presents a comparative study of active database systems and autonomous evolutionary information systems.

The rest of this paper is organized as follows: Section 2 and Section 3 give a brief introduction to active database systems and autonomous evolutionary information systems respectively, Section 4 comparatively discusses motivations, purposes, capabilities, and logical foundations of active database systems and autonomous evolutionary information systems. Section 5 shows that some facilities and their implementation techniques independently developed in active database systems and autonomous evolutionary information systems separately can be complementarily used each other and this will certainly result in progress of the both.

2. Active Database Systems

An ***active database system*** is a database system endowed with active rules and event monitoring scheme such that certain actions defined by active rules are automatically executed when certain events of data operations (e.g., inserted, deleted, updated, or selected) are detected and some particular conditions are met [20, 21, 27].

Active database systems enhance traditional database functionality with event-driven rule-processing capabilities, providing a flexible and powerful mechanism for advanced applications of database systems such as integrity constraints, views, authorization, statistics gathering, monitoring and alerting, and workflow management.

The goal and/or behavior of an active database is accomplished through its active rules. An active database system itself performs certain operations automatically in response to certain events occurring or certain conditions being satisfied. Active database systems allow their users and database administrators to define and specify the desired active behavior by means of active rules.

In general, ***active rules*** are typically expressed as Event-Condition-Action (ECA) rules of form as follows:

when <event> **if** <condition> **then** <action>

An active rule consists of the following three parts:

Event: causes the rule to be triggered

Condition: is checked when the rule is triggered

Action: is executed when the rule is triggered and its conditions is true

Whenever the specified event occurs, the rule is triggered and the corresponding action is executed if the condition is satisfied in the current database state. Active rules without the event part are called ***production rules***, and active rules without the condition part are referred to as ***triggers***.

Once a set of active rules is defined, the active database system monitors the relevant events. For each rule, if the rule's event occurs then the active database

system evaluates the rule's condition, and if the rule's condition is true then the active database system executes the rule's action, i.e., firing the rule.

Active database systems can efficiently perform functions that in passive database systems must be encoded in applications, e.g., general integrity constraints and triggers. Active database systems suggest and facilitate applications beyond the scope of passive database systems, e.g., data-intensive expert systems and workflow management. Active database systems can perform tasks that require special-purpose subsystems in passive database systems, e.g., simple integrity constraints, authorization, statistics gathering, and views.

From the viewpoint of logical foundation, active database systems are somehow based on classical mathematical logic. First, conditions of active rules are usually represented as formulas in classical mathematical logic. Second, the theoretical work on foundations of active databases is still rare and existing works are related the active rules to traditional production rules, logic programming, and deductive databases all of which are closely based on classical mathematical logic [3, 18, 19].

Historically, several active database features and proposals did appear in the 1970s and early 1980s. The DBTG language, defined by DBTG in the early 1970s, includes a mechanism for automatic procedure invocation in response to specified database operations. The screen-oriented query language QBE (Query-by-Example) for relational database systems, developed in the mid 1970s, includes a trigger facility for integrity constraint checking. A trigger subsystem was suggested for the System R relational database system research project in the late 1970s. The first paper motivating and describing a general active database system framework appeared in the early 1980s.

3. Autonomous Evolutionary Information Systems

An intelligent information system differs primarily from the traditional database and/or knowledge-base systems in that it can provide its users with not only data or knowledge stored in its database or knowledge-base by its developers and users, but also new knowledge, which are discovered or reasoned out automatically by the system itself from its database or knowledge-base. These new knowledge may be new propositions, new conditionals, new laws, new rules, and so on. Therefore, unlike a traditional information system serving just as a storehouse of data or knowledge and working passively according to queries or transactions explicitly issued by users and application programs, an intelligent information system serves as an autonomous and evolutionary partner of its users that discovers new knowledge by automated reasoning technique from its database or knowledge-base, cooperates with its users in solving problems actively by providing the users with advices, and has a certain mechanism to improve its own state of 'knowing' and ability of 'working'. Cheng named this type of information systems '***Autonomous Evolutionary Information Systems***' [9-11].

A ***reactive system*** is a computing system that maintains an ongoing interaction with its environment, as opposed to computing some final value on termination. An autonomous evolutionary information system is a reactive system with the capability of concurrently maintaining its database or knowledge base, discovering and providing new knowledge, interacting with and learning from its users and environment, and improving its own state of 'knowing' and ability of 'working'.

The term '***evolution***' means a gradual process in which something changes into a different and usually better, maturer, or more complete form. Therefore, the

autonomous evolution of a system, which may be either natural or artificial, should be a gradual process in which everything changes by conforming to the system's own laws only, and not subject to some higher ones. Note that in order to identify, observe, and then ultimately control any gradual process, it is indispensable to measure and monitor the behavior of that gradual process. ***Measuring*** the behavior of a computing system means capturing run-time information about the system through detecting attributes of some specified objects in the system in some way and then assigning numerical or symbolic values to the attributes in such a way as to describe the attributes according to clearly defined rules. ***Monitoring*** the behavior of a computing system means collecting and reporting run-time information about the system, which are captured by measuring the system. Measuring and monitoring mechanisms can be implemented in hardware technique, or software technique, or both. For any computing system, we can identify and observe its evolution, i.e., a gradual change process, only if we can certainly measure and monitor the system's behavior. Also, an autonomous evolutionary computing system must have some way to measure and monitor its own behavior by itself.

The most important core component of an autonomous evolutionary information system is a forward reasoning engine that must have the ability to generate new inference rules and discover new knowledge such that the new inference rules and knowledge can be used by the system itself to improve its own state of 'knowing' and ability of 'working' [5, 15].

From the viewpoint of logical foundation, autonomous evolutionary information systems are closely based on relevant logics [1, 2], in particular, strong relevant logics [8, 12] in the sense that only strong relevant logics can provide the logical validity criterion for reasoning rule generation [4, 5, 15].

In summary, autonomous evolutionary information systems have the following key characteristics and/or fundamental features: (1) An autonomous evolutionary information system is an information system that stores and manages structured data and/or formally represented knowledge. In this aspect, there is no intrinsic difference between an autonomous evolutionary information system and a traditional database or knowledge-base system. However, this characteristic distinguishes autonomous evolutionary information systems from those "intelligent systems" without a database or knowledge-base. (2) An autonomous evolutionary information system has the ability of reasoning to reason out new knowledge based on its database or knowledge-base autonomously. The ability of knowledge discovery is the most intrinsic characteristic of autonomous evolutionary information systems. (3) An autonomous evolutionary information system has the ability (mechanism) to improve its own ability of working in the sense that it can add new facts and knowledge into its database or knowledge-base autonomously and add new inference rules into its reasoning engine autonomously. It is in this sense that we say that the system is autonomously evolutionary. (4) An autonomous evolutionary information system communicates and cooperates with its users in solving problems actively by providing the users with advices and helps that are based on new knowledge reasoned out by itself, and therefore, it acts as an assistant of its users. (5) An autonomous evolutionary information system will ultimately grow up following its users such that two autonomous evolutionary information systems with the same primitive contents and ability may act very differently if they are used by different users during a certain long period of time. Therefore, the evolution of an autonomous evolutionary information

system is not only autonomous but also continuous such that its autonomous evolution is a persistently continuous process without stop of interaction with its user.

Historically, the first paper on the logical basis of intelligent information systems with the capability of reasoning rule generation appeared in 1992 [14]. The notion of autonomous evolutionary information system was proposed in 2000 [9, 10]. The necessity of self-measurement, self-monitoring, self-learning, and self-valuation, and continuous evolution to autonomous evolutionary information systems was shown in 2005 [6, 7, 11]. Cheng et al presented an autonomous evolutionary information system under development, named "HILBERT," for teaching and learning various formal logic systems underlying diverse reasoning forms in scientific discovery as well as everyday logical thinking [13].

4. Comparative Discussions

First of all, although with some similarity, the active database systems (ADSs for short) and autonomous evolutionary information systems (AEISs for short) were independently motivated by different ideas. The investigation of ADSs aims to enhance traditional database system functionality with event-driven rule-processing (trigger) capabilities to result in a more flexible and powerful formalism, while the investigation of AEISs aims to enhance traditional knowledge-based system functionality with reasoning rule generation capabilities such that a knowledge-based system can serve as an autonomous and evolutionary partner of its users that discovers new knowledge, cooperates with its users in solving problems actively, and has a certain mechanism to improve its own state of 'knowing' and ability of 'working'.

The difference of motivations between ADSs and AEISs leads to the difference between their purposes/goals in the sense that an ideal ADS should be a flexible and powerful database system with the ability to perform some tasks automatically according to conditions previously specified by its users, while an ideal AEIS should be an autonomous and evolutionary agent with the ability of self-growth. It seems that in the literature about ADSs there was few appearance of term "autonomous" but there was no appearance of term "evolution", while "autonomous" and "evolutionary" are two key characteristics of AEISs. Note that there is some difference between "automatic" and "autonomous" in the sense that the former usually means to do something according to some previously specified schedule, while the latter does not necessarily mean the existence of some schedule.

Because ADSs have no ability to generate active rules automatically, we can say that the capability of event-driven rule-processing is the most intrinsic characteristic of ADSs, while the capability of reasoning rule generation is the most intrinsic characteristic of AEISs. Therefore, form the viewpoint of evolution, the ADSs are still passive in the sense that they are not autonomous and evolutionary, while the AEISs are truly active in the sense that they have the ability to autonomously improve their own capabilities of 'knowing' and 'working'.

The differences between motivations, purposes, and goals of ADSs and AEISs leads to the most important difference between ADSs and AEISs, i.e. their different logical foundations. From the viewpoint of logical foundation, ADSs are closely based on classical mathematical logic [3, 18, 19], while AEISs are closely based on strong relevant logics [8, 12]. Because it is the logical foundation that intrinsically determines

future possible progress of ADSs and AEISs respectively, below we discuss this issue in detail.

Classical mathematical logic (CML for short) is a branch of mathematics. It was established in order to provide formal languages for describing the mathematical structures with which mathematicians work, and the methods of proof available to them; its principal aim is a precise and adequate understanding of the notion of mathematical proof. Both the object and the method of investigation of CML are mathematical proof. Given its mathematical method, it must be descriptive rather than prescriptive, and its description must be idealized. CML may be suitable to searching and describing a formal proof of a previously specified theorem, but not necessarily suitable to forming a new concept and discovering a new theorem because the aim, nature, and role of the CML is descriptive and non-predictive rather than prescriptive and predictive Therefore, if proposing a new question, inventing a new concept, or finding a new theorem is a thinking and discovering process, then we can say that discovery is not the object of investigation of CML, as logician Hao Wang said [26]: "It is a familiar misconception to believe that to do mathematical logic is to be engaged primarily in formal thinking. The important point is rather to make precise the concept of formal and thereby be able to reason mathematically about formal systems. And this adds a new dimension to mathematics."

In fact, because CML was established based on the classical account of validity (i.e., an argument is valid if and only if it is impossible for all its premises to be true while its conclusion is false), in the framework of CML, even if an argument is classically valid, the relevance between its premises and its conclusion cannot be guaranteed necessarily. On the other hand, because CML represents the notion of conditional, which is intrinsically intensional but not truth-functional, by the extensional truth-functional notion of material implication, if one regards the material implication as the notion of conditional and regards every logical theorem of CML as an entailment or valid reasoning form, then a great number of logical axioms and logical theorems of CML present some paradoxical properties and therefore they have been referred to in the literature as 'implicational paradoxes' [1, 2]. As a result, for a conclusion of a reasoning from a set P of premises based on CML, we cannot directly accept it as a correct and true conclusion in the sense of conditional, even if each of the given premises is regarded to be true and the conclusion can be regarded to be true in the sense of material implication.

The above two facts about CML result in that CML cannot underlie computational discovery and/or learning. First, because CML accepts the so-called principle of Explosion that everything follows from a contradiction, it is explosive but not paraconsistent. Second, CML adopts the classical account of validity and cannot underlie relevant reasoning; the validity in the sense of material implication is meaningless in the sense of conditional. Third, because CML adopts Modus Ponens for material implication (again, the notion of material implication is an extensional truth-functional notion) as its inference rule, a reasoning based on CML must be circular and/or tautological but not ampliative. Note that these facts are also true to those classical conservative extensions or non-classical alternatives of CML where the classical account of validity is adopted as the logical validity criterion and the notion of conditional is represented by material implication.

Consequently, any ADB cannot have the ability of discovery and/or learning, if it closely based on CML.

On the other hand, traditional relevant (relevance) logics were constructed during the 1950s in order to find a mathematically satisfactory way of grasping the elusive notion of relevance of antecedent to consequent in conditionals, and to obtain a notion of implication which is free from the so-called 'paradoxes' of material and strict implication [1, 2]. Some major traditional relevant logic systems are 'system E of entailment', 'system R of relevant implication', and 'system T of ticket entailment'. A major feature of the relevant logics is that they have a primitive intensional connective to represent the notion of conditional and their logical theorems include no implicational paradoxes. The underlying principle of these relevant logics is the relevance principle, i.e. for any entailment provable in E, R, or T, its antecedent and consequent must share a sentential variable. Variable-sharing is a formal notion designed to reflect the idea that there be a meaning-connection between the antecedent and consequent of an entailment. It is this relevance principle that excludes those implicational paradoxes from logical axioms and theorems of relevant logics. However, although the traditional relevant logics have rejected those implicational paradoxes, there still exist some logical axioms or theorems in the logics, which are not natural in the sense of conditional. Cheng named these logical axioms or theorems 'conjunction-implicational paradoxes' and 'disjunction-implicational paradoxes' [8]. Therefore, in the framework of any traditional relevant logics, even if a reasoning is relevantly valid, neither the truth of its conclusion in the sense of conditional nor the relevance between its premises and conclusion can be guaranteed necessarily. This situation is the same as that in CML.

In order to establish a satisfactory logic calculus of conditional to underlie relevant reasoning, Cheng has proposed some strong relevant (relevance) logics, named Rc, Ec, and Tc [8]. The logics require that the premises of a valid argument include no irrelevant and unnecessary conjuncts and the conclusion of that argument includes no irrelevant or unnecessary disjuncts. As a modification of traditional relevant logics R, E, and T, strong relevant logics Rc, Ec, and Tc rejects all conjunction-implicational paradoxes and disjunction-implicational paradoxes in R, E, and T, respectively. What underlies the strong relevant logics is the strong relevance principle: If A is a theorem of Rc, Ec, or Tc, then every sentential variable in A occurs at least once as an antecedent part and at least once as a consequent part.

At present, the strong relevant logics are best candidates for the fundamental logic to satisfactorily underlie computational discovery and/or learning. This proposition is based on the following facts. First, all relevant logics including strong relevant logics reject the principle of Explosion, and therefore, they are paraconsistent but not explosive. Second, the strong relevant logics provide a logical validity criterion that is relevant in the sense of strong relevance, i.e. for any valid reasoning based on a strong relevant logic, its premises include no irrelevant and unnecessary conjuncts and its conclusion includes no irrelevant or unnecessary disjuncts. Note that the logical validity criterion provided by traditional relevant logics is not relevant in this sense. The strong relevant logics are free of not only implicational paradoxes but also conjunction-implicational and disjunction-implicational paradoxes. Therefore, in the framework of strong relevant logic, if a reasoning is valid, then the strong relevance between its premises and its conclusion can be guaranteed necessarily, i.e. the logics can certainly underlie relevant reasoning in the sense of strong relevance. On the other hand, the logical validity criterion provided by strong relevant logics is truth-preserving in the sense of conditional. Note that the logical validity criterion provided by CML is truth-preserving only in the sense of material implication; it is not truth-preserving in

the sense of conditional. Also note that the logical validity criterion provided by traditional relevant logics is truth-preserving only in the sense of relevant implication; it is not truth-preserving in the sense of conditional. The strong relevant logics are free of not only implicational paradoxes but also conjunction-implicational and disjunction-implicational paradoxes. Therefore, in the framework of strong relevant logic, if a reasoning is valid, then the truth of its conclusion in the sense of conditional can be guaranteed necessarily, i.e. the logics can certainly underlie truth-preserving in the sense of conditional. Third, a reasoning based on any of relevant logics including strong relevant logics is ampliative but not circular and/or tautological. This is because the notion of entailment (conditional) that plays the most intrinsic role in any reasoning is represented in relevant logics by a primitive intensional connective satisfying the Wright-Geach-Smiley criterion, i.e. to come to know the truth of an entailment without coming to know the falsehood of its antecedent or the truth of consequent.

Consequently, it is the logical foundation of AEISs that makes AEISs can have the ability of discovery and/or learning as well as autonomous evolution.

5. Concluding Remarks

We have presented a comparative study of active database systems and autonomous evolutionary information systems. In conclusion, autonomous evolutionary information systems are intrinsically different from active database systems in purposes, capabilities, and logical foundations.

Active database systems have been well investigated and developed in a long time, while autonomous evolutionary information systems still in the early stage of investigation and development. Since both active database systems and autonomous evolutionary information systems are motivated by improving the passive characteristic of traditional database and/or knowledge-base systems, can we find some complementary development methodologies such that the facilities and their implementation techniques independently developed in active database systems and autonomous evolutionary information systems separately can be complementarily used each other?

First, the implementation techniques of event-driven rule-processing in active database systems may be useful to development of autonomous evolutionary information systems as well as development of epistemic programming language [25] for autonomous evolutionary information systems.

Second, the forward reasoning engine with general-purpose, FreeEnCal, may be used in active database systems as a inference rule generation engine, and the implementation techniques of forward reasoning may also be useful to rule-processing in active database systems.

Third, because the formal language of strong relevant logic is a conservative extension of the formal language of classical mathematical logic, all empirical data/knowledge (including empirical facts, empirical theorems, and inference rules) presented by classical mathematical logic can be easily transformed into empirical data/knowledge presented by strong relevant logic by replacing all material implication connectives with entailment connectives. Therefore, for any active deductive database system with a deduction engine based on classical mathematical logic or its various classical conservative extensions, if one transforms its data/knowledge by he formal language of strong relevant logic and replaces its deduction engine with a one based on

strong relevant logic (e.g., FreeEnCal), then the system can certainly manage data/knowledge for discovery and/or learning such that they can work well even if the active deductive database may include some inconsistency.

Finally, the logical foundations of autonomous evolutionary information systems can provide new theoretical direction for active database systems and therefore lead to development of some new powerful capabilities for them.

For any computational approach to scientific discovery and/or learning, flexible and powerful database/knowledge-base systems are indispensable [16, 17, 22-25]. Both active database systems and autonomous evolutionary information systems still need to improve their own capabilities of 'knowing' and 'working' in order to satisfy requirements from the advanced applications of computational scientific discovery and/or learning.

References

[1] A. R. Anderson, and N. D. Belnap Jr., *Entailment: The Logic of Relevance and Necessity, Vol. I*, Princeton University Press, Princeton, 1975.

[2] A. R. Anderson, N. D. Belnap Jr., and J. M. Dunn, Entailment: The Logic of Relevance and Necessity, Vol. II, Princeton University Press, Princeton, 1992.

[3] J. Bailey, G. Dong, and K. Ramamohanarao, K. (1998). "Decidability and Undecidability Results for the Termination Problem of Active Database Rules," *Proc. Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 264-273, ACM Press, 1998.

[4] J. Cheng, "Entailment Calculus as a Logical Tool for Reasoning Rule Generation and Verification," In: J. Liebowitz (Ed.), *Moving Towards Expert Systems Globally in the 21st Century -- Refereed papers accepted and presented at the 2nd World Congress on Expert Systems*, pp. 386-392, Cognizant Communication Co., Elmsford, NY, 1994.

[5] J. Cheng, "EnCal: An Automated Forward Deduction System for General-Purpose Entailment Calculus," In: N. Terashima and E. Altman (Eds.), *Advanced IT Tools, Proc. IFIP World Conference on Advanced IT Tools, IFIP 96 – 14th World Computer Congress*, pp. 507-514, Chapman & Hall, London, 1996.

[6] J. Cheng, "The Self-Measurement Principle: A Design Principle for Large-scale, Long-lived, and Highly Reliable Concurrent Systems," *Proc. 1998 IEEE-SMC Annual International Conference on Systems, Man, and Cybernetics, Vol. 4*, pp. 4010-4015, IEEE Systems, Man, and Cybernetics Society, 1998.

[7] J. Cheng, "Wholeness, Uncertainty, and Self-Measurement: Three Fundamental Principles in Concurrent Systems Engineering," In: *Proc. 13th International Conference on Systems Engineering*, pp. 7-12, 1999.

[8] J. Cheng, "A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery," In: E. Kawaguchi, H. Kangassalo, H. Jaakkola, and I. A. Hamid (Eds.), *Information Modelling and Knowledge Bases XI, Frontiers in Artificial Intelligence and Applications, Vol. 61*, pp. 136-159, IOS Press, Amsterdam, 2000.

[9] J. Cheng, "Temporal Relevant Logic as the Logical Basis of Autonomous Evolutionary Information Systems (Extended Abstract)," In: M. Khosrowpour (Ed.), *Challenges of Information Technology Management in the 21st Century, Proceedings of the 2000 Information Resources Management Association International Conference*, pp. 950-951, Idea Group Publishing, Anchorage, 2000.

[10] J. Cheng, "Autonomous Evolutionary Information Systems," *Wuhan University Journal of Natural Sciences, 6 (1-2), Special Issue: Proceedings of the International Software Engineering Symposium 2001*, pp. 333-339, Wuhan University Press, Wuhan, 2001.

[11] J. Cheng, "Autonomous and Continuous Evolution of Information Systems," In: R. Khosla, R. J. Howlett, and L. C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, 9th International Conference, KES 2005, Melbourne, Australia, September 14-16, 2005, Proceedings, Part I. Lecture Notes in Artificial Intelligence, Vol. 3681*, pp. 758-767, Springer-Verlag, Berlin, 2005.

[12] J. Cheng, "Strong Relevant Logic as the Universal Basis of Various Applied Logics for Knowledge Representation and Reasoning," In: Y. Kiyoki, J. Henno, H. Jaakkola, and H. Kangassalo (Eds.), *Information Modelling and Knowledge Bases XVII, Frontiers in Artificial Intelligence and Applications, Vol. 136*, pp. 310-320, IOS Press, Amsterdam, 2006.

[13] J. Cheng, N. Akimoto, Y. Goto, M. Koide, K. Nanashima, and S. Nara, "HILBERT: An Autonomous Evolutionary Information System for Teaching and Learning Logic," In: C. P. Constantinou and Z. C.

Zacharia (Eds.), *Computer Based Learning in Science, Conference Proceedings 2003, 6th International Conference, Vol. 1*, pp. 245-254, University of Cyprus, Nicosia, 2003.
[14] J. Cheng and Y. Kambayashi, “Entailment and Relevant Logics in Future Intelligent Information Systems,” In: *Proc. 2nd International Computer Science Conference*, pp. 71-77, IEEE Hong Kong Section, Hong Kong, 1992.
[15] J. Cheng, S. Nara, and Y. Goto, “FreeEnCal: A Forward Reasoning Engine with General-Purpose,” In: B. Apolloni, R. J. Howlett, and J. C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12-14, 2007, Proceedings, Part II, Lecture Notes in Artificial Intelligence, Vol. 4693*, pp. 444-452, Springer-Verlag, Berlin, 2007.
[16] J. Cheng, S. Nara, T. Koh, and Y. Goto, “The Theory Grid and Grid Theorists,” In: *Proc. 2nd International Conference on Semantics, Knowledge and Grid*, IEEE Computer Society, 2006.
[17] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow, *Scientific Discovery – Computational Explorations of the Creative Processes*, MIT Press, Cambridge, MA, 1987.
[18] G. Lausen, B. Ludascher, and W. May, “On Logical Foundations of Active Databases,” In: J. Chomicki and G. Saake (Eds.), *Logics for Databases and Information Systems*, pp. 389-422, Kluwer Academic, Dordrecht, 1998.
[19] I. Motakis and C. Zaniolo, “Composite Temporal Events in Active Database Rules: A Logic-oriented Approach,” In: Ling et al (Eds.), *Deductive and Object-Oriented Databases, Fourth International Conference, DOOD ’95, Singapore, December 4–7, 1995, Proceedings, Lecture Notes in Computer Science, Vol. 1013*, pp. 19-37, Springer-Verlag, Berlin, 1995.
[20] N. W. Paton (Ed.), *Active Rules in Database Systems*, Springer-Verlag, Berlin, 1987.
[21] N. W. Paton and O. Diaz, “Active Database Systems,” *ACM Computing Surveys, Vol. 31, No. 1*, pp. 63-103, ACM Press, 1999.
[22] J. Shrager and P. Langley, “Computational Approaches to Scientific Discovery,” In: J. Shrager and P. Langley (Eds.), *Computational Models of Scientific Discovery and Theory Formation*, pp. 1-25, Morgan Kaufmann, 1990.
[23] H. A. Simon, “Why Should Machines Learn?” In: R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach*, pp. 25-37, Springer-Verlag, Berlin, 1984.
[24] H. A. Simon, “What is a Systematic Method of Scientific Discovery?” In: *Systematic Methods of Scientific Discovery – Papers form the 1995 Spring Symposium, Technical Report SS-95-03*, pp.1-2, AAAI Press, 1995.
[25] I. Takahashi, S. Nara, Y. Goto, and J. Cheng, “EPLAS: An Epistemic Programming Language for All Scientists,” in Y. Shi, G. D. v. Albada, J. Dongarra, and P. M. A. Sloot (Eds.), *Computational Science - ICCS 2007: 7th International Conference, Beijing, China, May 27-30, 2007, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 4487*, pp. 406-413, Springer-Verlag, Berlin, 2007.
[26] H. Wang, *Popular Lectures on Mathematical Logic*, Van Nostrand Reinhold, 1981, Dover Publications, 1993.
[27] J. Widom and S. Ceri, “Introduction to Active Database Systems,” In: J. Widom and S. Ceri, (Eds.), *Active Database Systems – Triggers and Rules for Advanced Database Processing*, pp. 1-41, Morgan Kaufmann, 1996.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-173

Dynamic Agent Architectures for Complex Systems

Jeffrey Tweedale and Lakhmi Jain

School of Electrical and Information Engineering,
Knowledge Based Intelligent Engineering Systems Centre,
University of South Australia, Mawson Lakes, SA 5095, Australia.
{Jeffrey.Tweedale, Lakhmi.Jain}@unisa.edu.au

Abstract. Cognitive Science is a field of research attracting significant effort. This research requires the formation of teams of agents in order to dynamically configure the team with the ability to solve the decomposed task of the goal presented. Traditionally all tasks must be completed successfully or the team fails the goal [1,2]. A dynamic architecture would substitute agents within the team with alternative capabilities in order to succeed. It may even compromise and offer a partial solution and offer it to another system to complete. A good communications framework is required to pass messages between separate agent and other systems. Discussion about confined frameworks have recently been extended to enable individual students associated with our Knowledge-Based and Intelligent Information and Engineering Systems (KES) group to fast track the development of their research concepts. A *Plug 'n' Play* concept based on a multi-agent blackboard architecture forms the basis of this research. This paper highlights the core architecture, we believe is required for Multi-Agent System (MAS) developers achieve such flexibility. The research focuses on how agents can be teamed to provide the ability to adapt and dynamically organise the required functionality to automate in a team environment. The model is conceptual and is proposed initially as a blackboard model, where each element represents a block of functionality required to automate a process in order to complete a specific task. Discussion is limited to the formative work within the foundation layers of that framework.

Keywords: Autonomy, Architecture, Intelligent Agents.

1 Introduction

An exhaustive search on how the definition for an agent was derived [3]. Bratman [4], Russel and Norvig [5], Jennings and Wooldridge [6] all contributed to enhancing this definition, although many of the links need to be enunciated.

The features discussed include: autonomy (decisions/actions), reactivity (perception, behaviour or beliefs), pro-activeness (purposefulness/goals/ intentions) and social ability (ability to personify, communicate/coordinate/ collaborate, or interact). Each has been debated, for instance Frankcik and Fabian [7] classify gestures (written or spoken) as the ability to stimulate or *act*, Reynolds [8] and Tu [9] compare perception with the paradigm of *behaviour* and Bratman [4] attempts to tie this together into an architecture called Beliefs-Desires and Intentions (BDI). Evans [10] labels this approach as agency *orthodoxy of agent technology*, where intentions can be mapped to *opinions* and actions (re-actions) a consequence of *intentions* as implemented by Roa and Georgeff [11] in their BDI architecture.

A real-world systems takes inputs as sensors and react appropriately by modify the outputs as necessary. Simulation models rely on the same approach. They monitor sensors that stimulate decision making that may cause changes to outputs. Three architectures have been dominant in Artificial Intelligence (AI) research: blackboard systems, contract nets and frameworks. A blackboard may be thought of as a componentised system, where each box could function as separate concepts that represent the specified aspects of a system or sub-systems engaging the problem. This happens in an environment where experts are modular software subsystems, called knowledge sources, that are capable of representing different points of view, strategies, and knowledge formats, required to solve a problem. These problem-solving paradigms include: bayesian networks, genetic algorithms, rule-based systems, case-based systems, neural networks, fuzzy logic system, legacy (traditional procedural) software and hybrid systems.

2 Agent Architecture

Agent technologies, and in particular agent teaming, are increasingly being used to aid in the design of "intelligent" systems [2,12]. In the majority of the agent-based software currently being produced, the structure of agent teams have been reliant on that defined by the programmer or software engineer. The development of a model that extends the communications architecture of an agent framework that is adaptable when contacting a series of Multi-Agent System (MAS) or teams. The ideal properties of agents, includes: deliberative agents, reactive agents, interface agents (HCI) and mobile agents[1] [13]. Different systems may be instantiated with a variety of hierarchies, with each level performing predetermined tasks in a subordinate or supervisory role. An Agent Architecture is considered to include at least one agent that is independent or a reactive/proactive entity and is conceptually contains functions required for perception, reasoning and decision. The architecture specifies how the various parts of an agent can be assembled to accomplish a specific set of actions to achieve the systems goals. Wooldridge believes it is essential for an agent to have "the ability to interact with other agents via some communication language [14]". The question of how to classify agents has been debated since the first scripts originated. Some

[1] Inter and intra-net.

researchers preferred functionality, while others used utility or topology. Agreement on categories has not been rationalised, however three classes dominate. These classifications include: mobility, reasoning (such as reactive-deliberative), attributed models (such as those used in autonomy), planning, learning, and co-operative/collaborative/communicative agents [1]. Alternatively, Nwana chooses to classify agents using mobility, reasoning, learning, information and data fusion. Noting these differences, pressure is also emanating from within the Distributed Artificial Intelligence (DAI) community to include interaction within the beliefs, desires and intentions (BDI) agent paradigm with interaction into this definition.

A multi-agent system may be regarded as a group of ideal, rational, real-time agents interacting with one another to collectively achieve their goals. To achieve their goals, each one of these individual agents needs to be equipped to reason not only about the environment, but also about the behaviour of other agents. In this context, a strategy refers to a decision-making mechanism that provides a long term consideration for selecting actions to achieve specific goals. Each strategy differs in the way it tackles the solution space of the problem.

3 Agent Teams

The presence of multiple agents necessitates the need for coordination and the dynamic management of the functionality within a single component. This mechanism is responsible for the implementation of the agents' actions and also the planning process that goes into the implementation. Traditionally, depending on their approach to solving a problem of this nature, agents have been divided into three main categories: deliberative, reactive and hybrid systems. This problem escalates as the problems associated with co-ordination, especially in a dynamic environment, where the teams can switch roles at an instance. Forming agent teams generally requires prior knowledge of the resources and skills required by teams to complete a goal or set of decomposed tasks. This will be determined by the maturity and composition of the team, its members and any associated team. Each Agent may with other agents or group of agents (a Team). That agent may also interact or collaborate with other systems, within or across multiple environments. Agents can collaborate freely with other agents, although *Teams* are forced to communicate using traditional hierarchical methodologies. This structure needs to become more flexible in order to deliver efficiencies

4 The Dynamic Architecture

The maturity could be measured across each phase of *Team Development*' as described by Tuckman [15]. Personified characteristics/functions are possible in Beliefs, Desires, Intentions (BDI) agent architectures, although each consumes resources that may only be used by relatively few agents, many sporadically

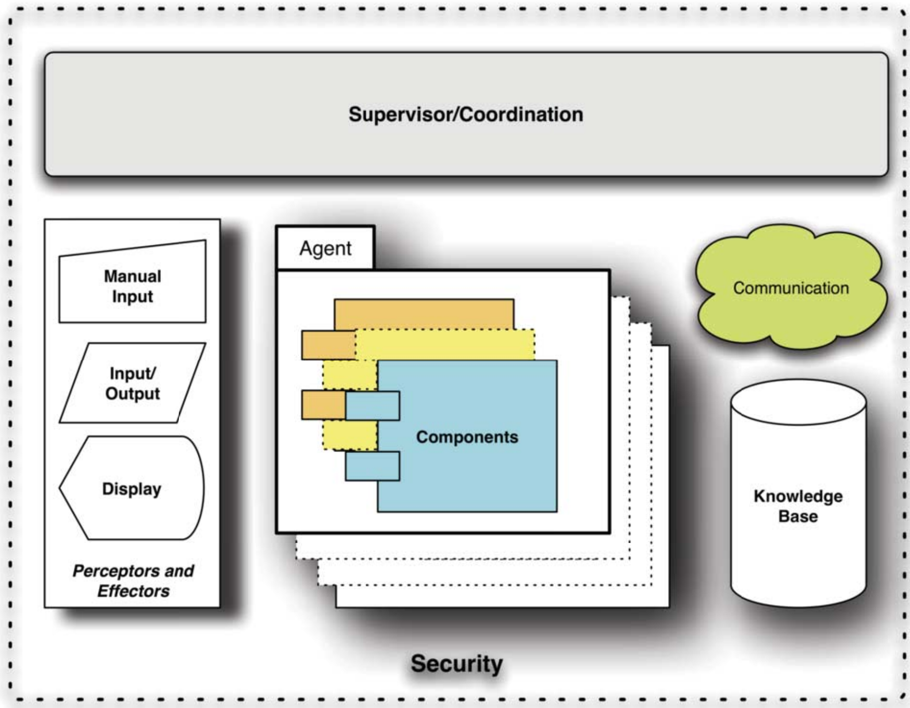


Fig. 1. Dynamic Agent Architecture Model

during its life. To reduce these overheads, only the resources/functionality required for a specified period would be instantiated and then released some time after that function is no longer required (Similar to the concept depicted in Figure 1). Based on this premise, interactions between agents within teams and between teams can generally be catalogued. We expect Intelligent Agents (IA) will retain their architectural foundations, but that the availability of more appropriate reasoning models and better design methodologies will assist in them being used increasingly in mainstream software development. Furthermore, better support for human-agent teams will see the development of a new class of intelligent decision support applications. Autonomous connectivity using one of the existing communication modes will be required to assist with the higher level interoperability.

Establishing communication between heterogeneous agent systems is crucial for building trust relationships between agent instances in these systems. Interchanging messages between different agent systems situated in different environments possibly affects all layers of the communication model, including transport protocol/system ports, ACL, semantic layers and Ontologies. The most common languages currently include: KIF, ACL, KQML and FIPA/ACL.

Incoming communication requests can be forked to autonomous threads capable of handling specific protocols and further processing and translation. To build a *proof of concept* for the communication layer of the TNC model we implemented a multi threaded client/server system to simulate the processing of multiple signal sources. These input stimuli are described by clients connecting to a server using a random protocol in our function model. The server thread models the communication interface of the receiving agent. The servers master thread accepts connection and forks it into a new thread specific for the determined protocol type, the received information is processed and committed to the knowledge base in the join step. The communication prototype is written in Java using Sun's^{TM} system library for multi-threading and networking. The function model is a simple Java application implementing the above architecture and visualizing the on-going communication.

5 Future Research

For the formation of a dynamic architecture that behaves with *Plug & Play* flexibility retains the power of embedded business logic with the flexibility of context switchable scenario data. Using a common interface with reflective components will provide minimal overheads to achieve the maximum efficiency available in a distributed environment. Establishing and implementing this process across a finite set of protocols will constitute the first phase of the implementation of the model. These agents must have the capability to autonomously context switch within a teaming environment to successfully achieve task/resource management in an MAS. Cohen [16] describes a Situation Specific Trust (SST) model that is comprised of a qualitative (informal core) that describes the mental attributes and includes a quantitative (prescriptive extension). Using taxonomy described it is easy to extend the capability of any system dynamically to solve both descriptive and ad-hoc problems. A concept demonstrator is being developed to demonstrate this research and enable further study into the dynamic operation of agent architectures, capable of auto negotiating with other agents, systems or sub-systems. This paper raises some of the initial issues raised and attempts to provide answers to the bottlenecks exhibited by pre-existing architectures.

Acknowledgments

We wish to express our greatest appreciation to the reviewers and referees for their visionary feedback, which were used to enhance the quality of this paper.

References

1. d'Inverno, M., Luck, M.: Understanding agent systems. Springer-Verlag New York, Inc., New York, NY, USA (2001)
2. Wooldridge, M., Jennings, N.R.: The cooperative problem-solving process. Journal of Logic and Computation **9**(4) (1999) 563–592

3. Franklin, S., Graesser, A.: Is it an agent, or just a program?: A taxonomy for autonomous agents. In: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages. Budapest, Hungary, Budapest, Hungary (1996) 193–206
4. Bratman, M.E.: Intentions Plans and Practical Reason. Center for the Study of Language and Information (1999)
5. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Second edn. Prentice-Hall, Inc., Englewood Cliffs, N.J. (2003)
6. Jennings, N.R., Wooldridge, M.: Applications of intelligent agents. In: Agent technology: foundations, applications, and markets, Secaucus, NJ, USA, Springer-Verlag New York, Inc. (1998) 3–28
7. Francik, J., Fabian, P.: Animating sign language in the real time. In: Applied Informatics, Innsbruck, Austria, 20th IASTED International Multi-Conference (2002) 276–281
8. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. Computer Graphics **21**(4) (1987) 25–34
9. Tu, X., Terzopoulos, D.: Artificial fishes: Physics, locomotion, perception, behavior. Computer Graphics **28**(Annual Conference Series) (1994) 43–50
10. Evans, R.: Varieties of learning. In Rabin, E., ed.: n AI Game Programming Wisdom. Volume 2. Charles River Media, Hingham MA (2002)
11. Rao, A., Georgeff, M.: BDI Agents: From theory to practice. In: Proceedings for the 1st International Conference on Multi-Agent Systems (ICMAS-95), AAAI Press, California (1995) 312–319
12. Urlings, P.: Teaming Human and Machine. PhD thesis, School of Electrical and Information Engineering, University of South Australia (2003)
13. Ehlert, P., Rothkrantz, L.: Intelligent agents in an adaptive cockpit environment. Technical Report DKE01-01, Version 0.2, Delft University of Technology, the Netherlands (2001)
14. Wooldridge, M., Jennings, N.R.: Theories, architectures, and languages: A survey, intelligent agents. In Wooldridge, M., Jennings, N.R., eds.: Lecture Notes in Artificial Intelligence. Volume 890. Springer-Verlag, Heidelberg (1995) 1–39
15. Mann, R.: Interpersonal styles and group development. American Journal of Psychology **81** (1970) 137–140
16. Cohen, M.S.: A situation specific model of trust to decision aids. Cognitive Technologies (2000)

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-179

Knowledge-base representation and reasoning for the autonomic management of pervasive healthcare

Eleftheria Katsiri,[1]
Birkbeck, University of London,
Malet Street,
London WC1E 7HX
Email: eli@dcs.bbk.ac.uk

Two recent programming paradigms, Body Area Networks (BAN) and Body Sensor Networks (BSNs) interact with a ubiquitous computing environment in order to monitor the health and well being of patients in hospitals or at home. Previous work in DSE, Imperial College, discusses the Self Managed Cell (SMC), as the basic architectural pattern of self-configuring and self-managing BAN networks. Devices in BANs become autonomic, by specifying their behavior reactively, in terms of the Ponder2 policy language. As a result, such networks require very little or no user input in order to adapt autonomously to changes in the users' environment, resulting from user activity, device failure, and the addition or loss of services.

Body Sensor Networks (BSN) are the next generation pervasive monitoring systems and are significantly lower-scale than BANs. BSN consist of very small nodes with limited computational memory and power resources that are not currently programmable in a reactive (rule-based) manner. This means that the SMC model cannot be applied directly to Body Sensor Networks; a new knowledge representation and reasoning model is needed to program such devices. This paper discusses the design decisions we took the design and implementation of an embedded policy interpreter (EPI) for autonomous BSN nodes.

---

[1] This work was completed while the author worked as a postdoctoral researcher at the Distributed Software Engineering group, Department of Computing, Imperial College London.

1. Autonomic Management of Healthcare systems

The authors previous work [amuse] substantiates the need for autonomy and self-management in body-area networks (BANs) that are used in health care, and particularly those that are used in order to monitor the long-term well-being of patients. Typically such networks consist of devices at the PDA level, small laptops and mainstream processors. Self-managed body area networks alleviate their programmers from the burden of configuring the multiple sensors and software components necessary for their operation. Furthermore, such systems require very little or no external input in order to adapt their behavior -so that they can remain reliable- according to the patient's context, the patient's clinical condition and interactions with other devices.

Another emerging networking paradigm is that where computationally-enabled nodes cooperate in forming a wireless Body Sensor Network (BSN). Body Sensor Networks are significantly smaller in scale than their counterparts, BANs. Body Sensor Networks consist of nodes that are themselves intelligent. Each node is effectively a low-power assembly of h/w components comprising of a sensor array, a micro controller unit (MCU) and a radio interface, connected through a common bus. BSNs are more structured than BANS and they have a high degree of homogeneity, as each node is usually implemented using a common embedded system architecture. This requires a different programming paradigm [bsn-policy].

It is easy to see that the behavior of each BSN node follows a common modus-operandi. It senses a physiological entity that is discretised in the form of samples. Whenever a sample that is ready to be processed in his input, the node reports its value back to the Master node. However, this common behavior can be programmed to be more resource-efficient; if only the significant or out-of-the-ordinary data is sent to the Master node, instead of each sample that is generated by the sensor, significant savings are achieved in the consumption of power. We note here that transmitting a data packet over low-power radio costs 10 times more than processing a data sample.

The above advocates for programmable node behaviors. As nodes have limited resources, it would be advantageous to be able to try and adapt the resource allocation to the current context of the node; the current context referring to the semantic “importance” of the data that is received by its associated sensors and the amount of available node resources. In existing BSN networks, programmable node behaviors are not currently possible in an efficient manner. Currently, the behavior of each node needs to be hard-coded and it forms part of the overall system policy. In order to be able to program node behaviors we need to be able to represent knowledge about their domain and reason with the properties of the knowledge predicates. The rest of the paper describes a knowledge representation and reasoning scheme that enables the programmability of BSN nodes thus enhancing their autonomy. Our work abides by the Ponder policy language as well the high-level principles of the Self-Managed Cell [amuse, bsn-policy].

The rest of the paper is organized as follows: Section 2 discusses the Ponder 2 policy service that is used in the SMC architecture. Section 3 presents our implementation of an embedded policy interpreter that realizes the common modus

operandi in BSN nodes. Section 4 discusses our embedded knowledge representation model, i.e. our representation of Ponder 2 policies in the restricted environment of a BSN node. Section 5 illustrates the use of the embedded policies in real-word pervasive healthcare scenarios.

2. The Ponder 2 Policy Service.

Policy-based management [ponder] is a proven, efficient autonomic management solution for the modus operandi of a BAN. An obligation policy is an Event-Condition-Action (ECA) rule specification that describes a set of event patterns (event specification) and conditions (condition specification) applied upon them, as well as the actions (action specification) to be triggered when the given situation is matched. For example the following policies could be specified for a BSN node consisting of sensors monitoring the recovery of a patient with a cardiac condition:

Policy 1
policy publishDataAtGivenPollTime(POLLTIME)
  on sensorF.timerE (POLLTIME) **do**
    sensorF.temperature (sensorF.readtempA ())

Policy 2
policy applyThresholdNormal
  on sensorF.heartRateE (hr) **do**
    if hr>MAXHRNormal then masterF.HeartRateE (hr)

Policy 3
policy applyThresholdActive
  on sensorF.heartRateE (hr) **do**
    if hr>MAXHRActive **then**
    masterF.HeartRateE (hr)

Policy 4
policy adaptThresholdToActivity
  on context(activity) do
    if activity == “running” then
      policies/applyThresholdNormal.disable();
        policies/applyThresholdActive.enable()

Policy 5
policy adapt RateToActivity
  on context(activity) do
    if activity == “running” **then**
      policies/publishDataAtGivenPollTime(RUNPOLLTIME);

Policy 6
auth+ /Master /os.{setfreq, setMinVal, stop, start}

Policy 7
auth+ /Master /policies.{load, delete, enable, disable}

The first policy is an event registration policy. It causes the BSN node to register for events published by its own temperature sensor, at a given polling rate that is specified with the value POLLTIME. Note that this type of event registration specification is slightly different form traditional distributed systems, where the event client has no control over the rate at which events are published (is this right? Or should we talk about OUT values). However, the proposed technique is appropriate for sensor networks, where the sensing is continuous and it needs to be sent at different time intervals depending upon the application and the sensor. A policy that determines the sampling rate is hence appropriate as part of the event registration process.

The second policy is triggered by a heart rate (HeartRateE) event and it publishes an external event at the Master node, whenever the value of the heart rate exceeds a threshold that is appropriate for normal behavior. The third policy is similar, but it only publishes an event at the Master node when the value of the received heart rate event exceeds a higher threshold that is associated with the normal condition while running. Policy 4 enables Policy 3 and disables policy 2 whenever the users is known to be running and Policy 5 adapts the polling rate to a desirable predefined rate, whenever the user is running. Policies 6 and 7 are authorisation policies and permit management of the oxygen saturation monitoring and the policies.

2.1. Roles

A role [domains] is a specification that groups setup and obligation policies on behalf of a Master node.

A role registration specification includes an event specification with temporal properties that contains all the information that the server needs in order to subscribe to events according to the specified temporal data. For example, the following registration is part of the role *tempRoleT* that defines the behavior of a temperature sensor on behalf of the Master node. It is equivalent to a subscription to events of type sensorF.Temperature at a rate of POLLTIME, for a maximum period of MAXTIME.

```
role tempRoleT(masterF, sensorF, POLLTIME, MAXTIME)
   on sensorF.timerE (POLLTIME) do
    sensorF.temperature (sensorF.readtempA ())

   on sensorF.temperatureE (temp) do
      if temp>MAXTIME then masterF.temperatureE (temp)
```

Role registration is two-fold: it is responsible for setting up the server for subscription to the event specification and it is responsible for storing in memory the policies that specify what actions need to be taken in response to the received event. For example, the above registration specification is equivalent to the following sequence of policy specifications:

policy pollSensor(POLLRATE)
on policyAdded (pollSensor) **do**
sensorF.setTimerE (POLLRATE);

policy publishData()
on sensorF.TimerE.fired() **do**
SensorF.getData ();

3. An embedded Policy Interpreter

Policies such as the ones discussed here can not be applied to an existing BSN node, in real-time. Currently, the same functionality would have to be hard-coded. For this reason we designed the embedded Policy Interpreter (EPI) [bsn-policy]. The EPI is a reactive system that is responsible for accepting, matching and managing roles and obligation policies on behalf of a client application that we refer to as a Master node.

The EPI works on the following algorithm: each time an event is received, the set of existing policies that are stored in an internal data structure are evaluated in order to determine whether the event that has been received satisfies the event specification part of the policy. We refer to the successful evaluation of a policy as a policy match or a matching cycle. As result of a policy match the EPI triggers one of the specified actions provided by the server, e.g.Toggle the red LED, send a packet to the master node, set a timer, and perform a remote invocation. A list of available actions is shown in Table 1. The algorithm that is performed by the EPI is shown in Figure 1. In Figure 1, the transition labeled DataReady*(type, value)* represents the asynchronous receipt of an event of type *type* and value *value*. For example, this may correspond to a reading of a temperature sensor.

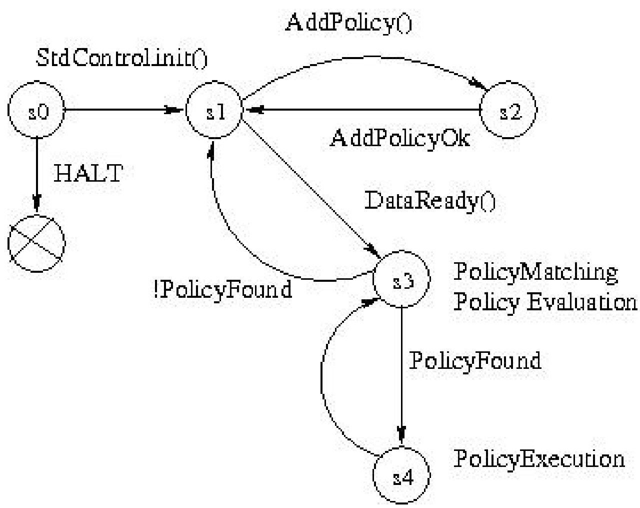


Figure 1: The EPI algorithm encoded as a finite state machine.

Role registrations can also be inputs to the EPI. We mentioned before that a role registration is equivalent to a set of policies. Whenever a role registration is added in the system, all the policies that comprise it, are evaluated in order. If a setup policy is found, then after it has been successfully evaluated it is executed e.g. setting up appropriate polling rates for sensors. Any obligation policy that follows as part of the role registration is stored in memory where it will be evaluated in the next policy match cycle. The EPI runs according to this algorithm (Figure 1) until it is manually halted.

3.1. Implementation

The design and implementation of the EPI caters both its *limited computational resources* as well as for its *real-time requirements.* BSN nodes typically run embedded software such as tinyOS. In our prototype we are using tinyos and nesC. Both the policy and the event model are implemented as a C structs. The main data-structure that stores the policies is implemented as a fixed-size array (20 policies) of policy structs. This is an efficient solution in terms of space. The implementation of the actions is in the form of C functions, embedded in a single nesC component. This is not the most flexible solution as actions cannot be added without modifying the EPI nesC code, but we feel that there is limited need for new action specifications in such as restricted environment.

4. An Embedded Event –Condition-Action policy model

For Biomedical SMC systems we have defined the following policy structure, encoded as a tuple:
]***Policy*** *<event_name, [params], p_id, p_state, operator, opearand, action_code, [action_params] >*

In the EPI, an event is modeled as a *type-value* pair. However, we have catered for a fixed number of event parameters. We have defined the events shown in Table 2.

Events	**Code**
Sensors.AccelX.DataReady	10
Sensors.AccelX.TimerE	11
Sensors.AccelY.DataReady	20
Sensors.AccelY.TimerE	21
Sensors.Temp.DataReady	30
Sensors.Temp.TimerE	31
Receive(originator, msg)	50

Table 1: Event Specification

A simple event specification is the one that models a single event occurrence, as shown in the policy definition above. The field *p_id* refers to an integer that identifies uniquely each policy. The field *pstate* can take the values 1, 2, 3 where 1 stands for "active", 2 for "inactive" and 3 for setup policy. The condition part of the policy model is specified with an *operator* and an *operand*. The operator field takes the values {1, 2, 3} standing for {=, <,>} respectively. The *action code* of the policy takes a value form those shown in Table 1.

Action	***Code***
Sensors.Temp.getData	100
Sensors.Temp.setSamplingRate	120
Sensors.Accel.getData	140
Sensors.Accel.setSamplingRate	160
LEDs.Red.On	201
LEDs.Red.Off	221
LEDs.Green.On	202
LEDs.Green.Off	222
LEDs.Blue.On	203
LEDs.Blue.Off	223
LEDs.Red.Toggle	501
LEDs.Green.Toggle	502
LEDs.Blue.Toggle	503
LEDs.Red.StartBlinking	560
LEDs.Green.StartBlinking	580
LEDs.Blue.StartBlinking, blue	590
LEDs.Red.StopBlinking	600
LEDs.Green.StopBlinking	620
LEDs.Blue.StopBlinking	640
Send Msg(msg, destination)	701
Timer.Start	702
Timer.Stop	800
Timer.SetFrequency	900

Table 2: Action Specification

5. Example: Role Registration.

We have defined 3 types of policies: *obligation* and *authorisation* policies, as in the SMC model, and also *setup* policies that represent actions. Such actions are used for example during event registration for setting up timers in order to adapt polling rates. For example, consider the role:

role tempRoleT(masterF, sensorF, POLLTIME, MAXTIME)
  on sensorF.timerE (POLLTIME) **do**
   sensorF.temperature (sensorF.readtempA ())

  on sensorF.temperatureE (temp) **do**
   if temp>MAXTIME then masterF.temperatureE (temp)

We mentioned before that the above specification is equivalent to the following set of policies:

pollSensor(POLLRATE)
  on policyAdded (pollSensor) **do**
   sensorF.setTimerE (POLLRATE);

policy publishData()
  on sensorF.TimerE.fired() **do**
   SensorF.getData ();

The first policy may be identified as a “setup” policy in which case it will interpreted as follows: the specification *sensorF.setTimer(POLLRATE)* causes setting and starting the timer *(TimerE)*, which controls the sampling of sensor sensorF, at the desired rate *POLLRATE*. Each time *TimerE* fires, the internal event *sensorF.TimerE.fired ()* is published.

Using the ECA model we presented above the setup policy is modeled with the following tuple:

Policy<”Sensors.Temp.timerFired”, 0, 0, 0, 101,”setup”, 0, 0, 0, 0, 0, 0>

Substituting each event and action specification with its associated code (as shown in Tables 1 & 2) and assuming MAXTEMP has a value of, say, 36, we get a more concise policy representation:

Policy<31, 0, 0, 0, 101, 0, 0, 3, 0, 0, 0, 0>

The second policy, publishData() is identified as an “obligation policy” is stored in memory directly following the role registration. It dictates that whenever the timer of the temperature sensor fires, the data should be made available to the EPI by the appropriate data source, i.e. the temperature sensor.

Taking into consideration the internal implementation of tinyOS, the above is equivalent to the following:

The *event sensorF.TimerE.fired ()* causes the invocation of *sensorF.getData ()*which requests the data from the sensor. In our implementation of the EPI all getData() operations trigger a *dataReady (struct event *e)* event, after the data has been acquired and is available in the server.

This policy can also be converted to the ECA model presented here:

Policy<”Sensors.Temp.timerFired”, 0, 0, 0, 102,”active”, 0, 0, Sensors.Temp.getData, 0, 0, 0>

Substituting each event and action specification with its associated code (as shown in Tables 1 & 2) and assuming MAXTEMP has a value of, say, 36, we get a more concise policy representation:

Policy<31, 0, 0, 0, 102, 1 0, 0, 100, 0, 0, 0>

Note that although the event name is the same in the specification of the first and second policies, they are interpreted differently, because the first one is a setup policy while the second one is an obligation (active) policy. This is a feature of the EPI.

A third policy in the role registration defines the action that needs to take place whenever a*dataReady (struct event *e)* event occurs.

policy applyThreshold
    on Sensors.Temp.dataReady (int value) **do**
        if value >MAXTEMP then Send (master, value);

For example, the above policy dictates that whenever an event measurement occurs (in the form of a received event), the measurement should be forwarded to the Master node only when the temperature exceeds a threshold defined by MAXTEMP. Using the ponder notation we can write:

Using the model we presented in this section, the above policy can be written as:

Policy<”Sensors.Temp.dataReady”, value, 0, 0, 103,”active”, “>”, MAXTEMP, “Send”, 0, 0, 0>

Substituting each event and action specification with its associated code (as shown in Tables 1 & 2) and assuming MAXTEMP has a value of, say, 36, we get a more concise policy representation:

Policy<10, 0, 0, 0, 101, 1, 3, 36, 701, 0, 0, 0>

6. Managed Policy Objects.

Using the above methodology, we have also provided embedded knowledge representations for managing roles and policies. In order to be able to create invocations on a policy or a role, an object-wrapper is needed:

```
result_t PolicyWrapper(struct policy *p, char command[2]){
     case (command){
          "en": result=enablePolicy(uint16_t Id);break;
          "di": result=disablePolicy(uint16_t Id);break;
          "ex": result=executePolicy(uint16_t Id);break;
          "ev": result=evaluatePolicy(uint16_t Id);break;
          "ec": result=evaluateCondition(uint16_t Id, struct event_t *ev);break;
          "gn":result=getPolicyByName(struct policy *p);break;
          }
          return result;
}
```

```
policyObjectWrapper{
getPolicyByName(int Id)
enable(int id);
disable(int id);
removePolicy(int id);
execute(int id );
evaluate(int id, struct event_ t *ev);
}
```

Figure 2 shows the nesC interfaces on the Policy Interpreter nesC component that allow the management of policies. The policy script control implements our compiler; given a policy in our knowledge representation the compiler invokes the appropriate interface on the Policy interpreter.

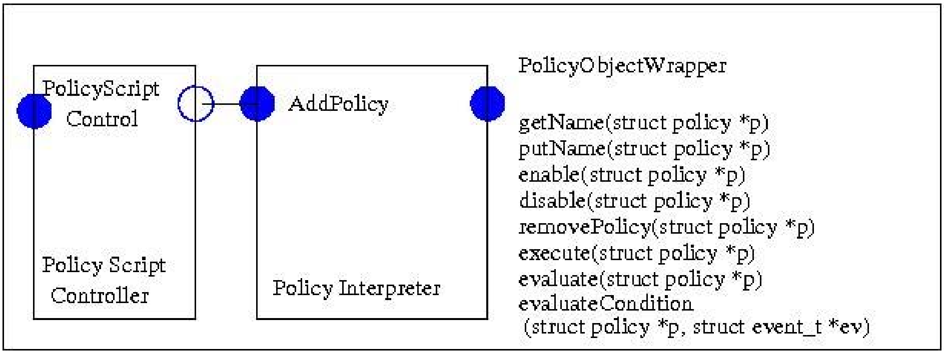


Figure 2: The policyObjectWrapper and Interface

7. Conclusions – Future Work

We are also considering a second version of the EPI that is more modular and therefore flexible in terms of adding event specifications and action implementations. In the next release of the EPI the event and action specification will range in the lower integer scale (e.g. event codes from 1 to 11) and action codes from 1 to 20) in order to reduce the overhead in numerical processing.
We are also working in the direction of enhancing the security of the the EPI using techniques such as an access-control list [bsn-access] and role-based access control.

References:

[ponder] N. Damianou, N.Dulay, E. Lupu, M.Sloman. Ponder: A language for specifying Security and Management Policies for Distributed Systems.

[bsn-policy] S.L. Keoh, K. Twidle, N. Pryce, A.E. Shaeffer-Filho, E. Lupu, N. Dulay, M.Sloman, S. Heeps, S. Strowes, J. Sventek, and E. Katsiri. Policy-based management for body sensor networks. In Proceedings of the 4th International Workshop on Wearable and Implantable Body Sensor Networks , Aachen, March 2007.

[bsn-access] E. Katsiri. An Access Control Model for Body Sensor Networks using authorisation policies. To appear in the 5th International workshop on Werable and Implantable Body Sensor Networks, Hong-Kong, June 2008

[amuse] E. Lupu, N.Dulay, M.Sloman, J. Sventek, S. Heeps, S. Strowes, S.-L. Keoh, A. Schaeffer-Filho, and K. Twidle. Amuse: Autonomic manage-
ment of ubiquitous e-health systems, 1994.

[domains] M. Sloman and K. Twidle. Domains: A framework for structuring management policy, 1994.

This page intentionally left blank

Intelligent User Interfaces and Human-Machine Interaction

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-193

Personalized content-based retrieval in mobile music services

Paraskevi S. Lampropoulou and Aristomenis S. Lampropoulos and George A. Tsihrintzis
Department of Informatics
University of Piraeus
Piraeus 185 34, Greece.
e-mail: {vlamp,arislamp,geoatsi}@unipi.gr

Abstract. ALIMOS is a middleware system that facilitates the access to digital music libraries in push technology-based mobile services. Specifically, a mobile user is provided with the ability to query for music files that belong to the same genre by simply sending an example music file from his/her mobile device. The personalization in ALIMOS is based on information collected while the service is used by a user and provides the user with results that have a higher content similarity with the query. The personalization mechanism is based on a novel application of clustering algorithms.

Keywords. Mobile Service, Content-Based Retrieval, Personalization

1. Introduction

An expanding area of mobile services over the last few years is the area of multimedia services. Mobile multimedia services provide means for delivering multimedia content and information between two mobile stations or between a mobile station and a mobile operator. Mobile multimedia services give users the ability to use their device for entertaining purposes more than for telecommunication purposes. Many services for handling, searching, and sharing images, audio, and video were first created to cover internet user needs. However recently are highly required internet-based multimedia services to become available through a mobile environment. The rapid growth of multimedia mobile services is further boosted by the evolution in mobile device technologies. Indeed, modern mobile devices have inherent abilities to provide multimedia services as they are equipped with camera, radio, music players and other modules that create and handle multimedia data. Moreover, a serious limitation of mobile devices, that is, their limited storage memory, was eliminated in the recent years. These advances have led users to consider their mobile devices not only as simple communication devices, but rather as multimedia entertainment and storage equipment [1]. Moreover modern mobile devices and mobile networks allow their users to place video calls or to exchange messages of multimedia content [2, 3]. However, these advances do not guarantee personalization of multimedia retrieval services if knowledge-based techniques are not used.

Multimedia information retrieval with use of mobile devices is an emerging research area. More specifically, mobile music services, which are not as developed as image or

video services, can benefit a lot from research into the development of advanced applications that implement content-based retrieval from music databases and collections. Because of a significant increase in the availability of content in mobile devices, the user sometimes devotes significant amounts of time and effort in searching and downloading his/her preferred content. Thus, there is an imposing need for mobile multimedia services which allow the user to search and deliver multimedia content to mobile devices in an effective, efficient, and personalized manner.

The need to develop services that satisfy the individualities of their users has led to the personalization of services [4]. The personalization of a service attempts to adapt the service to the needs and preferences of a specific user or a group of users. The most critical steps to personalization of a service are (1) the identification of the necessary information about the user or group of users, (2) the mechanism of collecting this information, and (3) the strategy through which this information is utilized by the service to provide personalized content delivery to the service users. The main benefit from personalization of a service is significant reduction of the amount of unnecessary information sent to the service users and the time a user spends to receive the content he/she desires [5].

In the area of mobile services, there are many directions and factors that need to be considered to personalize a service. One direction leads to personalization based on adapting the mobile service to the capabilities of the mobile device and telecommunication operator [6]. Another direction leads to personalization of mobile services based on the location of the user and use of global positioning retrieval systems (GPRS) [7]. That is, a mobile server provides information to the user, which is dependent on his/her location. Finally, another direction leads to personalization based on user preferences and data collected while the user uses the mobile service [8].

The latter form of personalization is used to enhance several mobile services for searching and downloading multimedia content, in general, or audio files, in particular. These services are developed on top of middleware systems which allow information exchange between modules that perform database search and modules that handle the mobile communication tasks. Recently, we developed and evaluated a middleware system [9, 10], called ALIMOS,[1] which we enhance with a personalization mechanism in the present work. As a middleware system, ALIMOS facilitates the access to digital audio/music libraries in push technology-based mobile services. Specifically, our system provides a semi-automatic interface that allows mobile users to interact with a digital audio/music library and find/retrieve audio/music files in a flexible way that combines mobile technologies and content-based retrieval techniques. Thus, a mobile user is provided with the ability to query for audio/music files that belong to the same genre by simply sending an example music file from his/her mobile device. The personalization in ALIMOS is based on the information collected through the use of service from the user in order to provide the user with results that have a larger content similarity with the one input file.

The paper is organized as follows: Section 2 reviews related works, while Section 3 presents an overview of ALIMOS as it has been developed. Section 4 describes the personalization technique incorporated into the back-end of ALIMOS. Finally, conclusions are drawn and future research directions are illustrated in Section 5.

---

[1] ALIMOS stands for *A*ccessing digital *LI*braries in *MO*bile *S*ervices.

2. Related Work

In this section, we review briefly the most relevant works on personalization of mobile services: One approach, called *mobileMM4U*, is a software engineering framework for providing applications with the necessary components to dynamically generate personalized multimedia content for mobile devices [11]. Based on mobileMM4U, an approach was presented in [7] of a generic tourist guide which integrates the support for creating personalized multimedia presentation with mobile location-based services. Depending on the availability of a wireless network, a user can go on a trip with pre-generated personalized presentations, but have them generate on demand while being on his/her way. A user can use this service to pre-plan and pack potentially relevant multimedia information on his/her mobile device. Additionally, post processing of the tour is allowed, such as the trip path can be logged and annotated with pictures taken with the userŠs mobile digital camera.

Another approach to providing personalized multimedia content is that proposed by [6]. In this work, they introduce a method of recommendation of personalized contents according to preference clones using a collaborative filtering technique in a mobile environment. This method divides the user group to two sub-groups by analyzing the degree of match of preferences of members of the sub-groups. The division process is applied recursively to each sub-group and organizes the sub-groups into a binary decision tree (BDT).

A third approach based on content browsed by users for creating user profiles is that proposed in [12]. In this work, a service platform, called *iMobile*, transcodes video content based on user and device profiles. This approach addresses various issues including authentication, authorization, transcoding, adaptation, and deployment.

3. ALIMOS Overiew

ALIMOS is a *middleware* tool that allows a mobile user to submit a query to a digital music library and retrieve music files that belong to the same musical genre ("are similar to the query") by sending an example music file from his/her mobile device [9, 10]. ALIMOS was developed to alleviate the limitations of existing mobile Content-Based Music Retrieval(CBRM) systems. Specifically, ALIMOS does not require an operating system in the mobile device, nor do any APIs or specific client need to be installed. ALIMOS uses Push technology which is supported by all mobile devices. Additionally, the CBMR procedure followed in ALIMOS is quite simple. Indeed, the user initiates the service by sending a Multimedia Mobile Service(MMS) to the server containing the audio file, similar music files to which he/she is interested in retrieving.

The response of ALIMOS is not *content delivery* that may not be the desired result and would, thus, result in unnecessarily high costs to the user. Instead, ALIMOS returns a simple push message containing a list of appropriate links. The links are named so that the user can decide from the link name whether to proceed to selective downloading of several of these music files from the web server to his mobile device. In our application, the corresponding ALIMOS architecture is illustrated in Fig. 1 as a multi-tier application with a *front-end* and a *back-end* level.

Specifically, the front-end level includes all those modules that implement the communication between the user, i.e., the mobile network and the application, while the back-

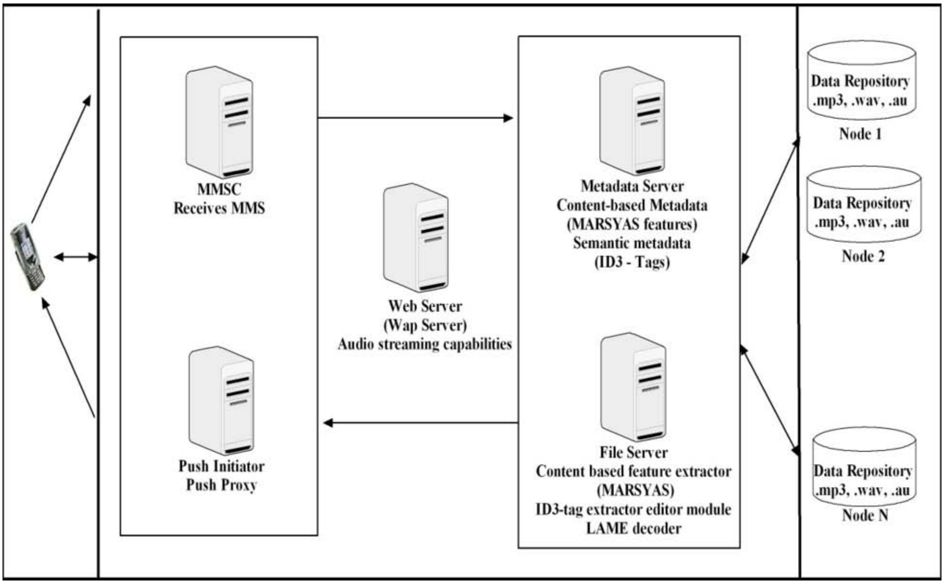


Figure 1. ALIMOS Architecture Overview

end level refers to all those modules that implement the content-based retrieval mechanism. Even though the music databases may be distributed and separate from the CBMR server in the back-end and the Push Proxy Gateway (PPG), the Push Initiator (PI) and the Wireless Access Protocol (WAP) servers in the front-end, the design of ALIMOS is such that these various modules are not visible to the users and all communication needs among the modules are handled automatically by ALIMOS.

The back-end level of ALIMOS consists of four modules. The first module is a *feature extractor* that captures low-level audio features [13]. The second module is an *id3 tag extractor* and an interface for the user to update the extracted semantic meta-data and add other possible semantic keywords. In ALIMOS, we follow the framework developed for *image* retrieval in [14], which unifies semantics an d content-based features. Thus, we allow user-supplied relevance feedback on both music semantics keywords (herein artist name, genre, etc.) and the music content-based features to learn semantic relevance between image clusters and improve the retrieval performance.

The third module realizes the process of various *clustering* methodologies, including fuzzy c-means, spectral, hierarchical, and artificial immune network-based clustering [15]. Clustering is used for the initialization of ALIMOS and applied on low-level content-based features to reveal the predominant classes in the music database. The fourth module realizes the *retrieval process*. The application makes a query by submitting an example which has been sent by the user with a MMS. The search for music pieces similar to the user example is implemented in the restricted format sense, in which either the Euclidean or the cosine distance metric are used to find data that are globally similar to the query. No attention is paid to the query scale. In the end, a result list is returned of the relevant music files in the library, found on the basis of their low-level (content-based) similarity to the query matching files.

The ALIMOS operation can be described by the following steps which can be diagrammatically presented in the in Fig. 2.

1. The user submits a MMS containing the query audio file
2. The server receives the MMS, extracts the query audio file and stores the necessary user information (e.g., sender's telephone number).
3. The server submits the audio file to the content-based retrieval (CBR) sub-system of the back-end level.
4. The CBR sub-system of the back-end level returns to the PPG a list of the four most relevant results.
5. The PPG forwards this list to the user in the form of a push message. The results in the push message are not the actual audio files, but rather links to the WAP server named after the music piece title and performing artist name. An example of an answer of ALIMOS is illustrated in Fig. 3.
6. The user may download from the WAP server the audio file(s) that he/she selects.

In the example we present, the user sends a MMS by selecting the "Led Zeppelin-All My Love" audio file from the music file folder of his mobile. The user expects a reply containing the results of the search for songs of the same musical genre. The user receives a message with the results of the system. The results are links to the music files in the WAP server. The name of each file consists of two parts. The first part is the name of the performing artist or group, while the second part is the title of the song. The ALIMOS results for the "Led Zeppelin-All My Love" query are the following: "Led

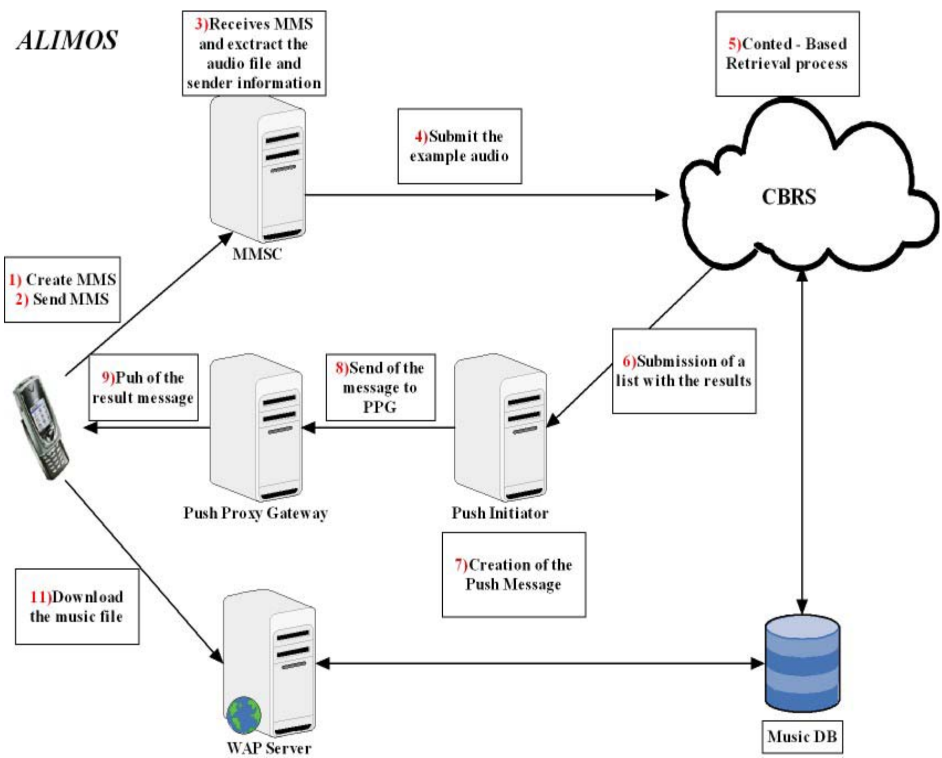


Figure 2. Step Flow during ALIMOS Operation

Figure 3. ALIMOS Reply to User Query = "Led Zeppelin-All My Love"

Zeppelin- In The Evening", "Pink Floyd- Money", "Pink Floyd- The Dogs Of War", "U2 - Whose gonna ride your wild horses", which all belong to the same genre of rock music.

4. Personalization In ALIMOS

As we described in the previous section, the back-end level of ALIMOS consists of four modules. In this paper, the back end of our middleware system is complemented with a new module which enhances ALIMOS with a personalization mechanism. The goal of the personalization mechanism is to allow the service to return audio/music search results that are not only similar in content and of the same genre as the query audio/music file, but have also taken into account user preferences as identified from previous use of the service. Specifically, the personalization mechanism is based on user information as described the following.

Each user is identified by his/her mobile number. Each time a user submits a query audio/music file (in the form of a MMS), information is collected in the back-end which consists of the music genre of the submitted audio file and the audio file or files downloaded by the user from the results returned to him/her via a push message sent by ALIMOS. This mechanism, creates a user profile of his/her preferences in terms of music genre, which is used for more personalized future retrievals.

Specifically, the user profile is created by training an ensemble of RBF-SVM classifiers, which are later fine-tuned via the information about the files downloaded by the user from those returned by ALIMOS in response to previous queries by the same user. In steps, the procedure is as follows:

1. The user seeds the search, by submitting a query (target music piece). The feature extractor extracts a set of values for a corresponding set of 30 audio features.
2. Each classifier returns the *confidence* of the query file. The decision value (confidence) is the distance of the query from the margin of the specific class. The class is selected which corresponds to the classifier with the highest decision value.
3. The selected classifier returns 5 songs with similar decision value from the margin of the specific class.

Table 1. Query File: "Led Zeppelin-All My Love"

Title	Decision Value
1.Led Zeppelin - In The Evening	0.99
2.Pink Floyd - Money	0.97
3.Led Zeppelin -The Rain Song	0.90
4.Pink Floyd- The Dogs Of War	**0.89**
5.U2 - Whose gonna ride your wild horses	0.86

Table 2. Query File : Pink Floyd- The Dogs Of War

Title	Decision Value
1.U2 - Running to stand still	0.88
2.Led Zeppelin -In The Evening	0.86
3.Pink Floyd - Nobody Home	0.77
4.Pink Floyd - Run Like Hell	0.75
5.U2 - I Will Follow	0.72

4. If the user selects to download a music/audio from the results of previous search using ALIMOS, the system updates the query using the confidence value of the selected song and searches for songs which are close to the new decision value.

In this process, the user preference is captured and stored using the class ID and the latest decision value from the margin of the specific class. This allows the adjustment of the user preference according to a specific class, while attention is paid to a sub-space of this class according to the decision value from the margin of the class.

An example of the personalization process is presented below. In the initial step the user submits the query. The query corresponds to the file "Led Zeppelin-All My Love". This file is processed by the content-based retrieval process in the back-end level of ALIMOS. This file is classified to the "Rock" class. As it is the first time the user submits a query the system returns five songs from the selected class with the highest decision value. This is presented in Table 1. The ALIMOS return the results of selected five songs as a push message as explained in Section 3. In the next step, the user downloads from the results the song "Pink Floyd- The Dogs Of War" which its decision value is 0.89. Then the system of ALIMOS stores this information to the users profile as a feedback for following retrievals from the specific class. The system uses this information to retrieve songs with decision value near to the value 0.89 as presented in Table 2.

5. Conclusions, and Future Work

Our work develops a middleware system called ALIMOS. ALIMOS facilitates the access to digital audio/music libraries in push technology-based mobile services. Specifically, our system provides a semi-automatic interface that allows mobile users to interact with a digital audio/music library and find/retrieve audio/music files in a flexible way that combines mobile technologies and content-based retrieval techniques. Thus, a mobile user is provided with the ability to query for audio/music files that belong to the same genre by simply sending an example music file from his/her mobile device. In this paper, previous versions of ALIMOS are enhanced with personalization capabilities.

Future work in this area will follow the directions of further evaluation of the personalization mechanism of ALIMOS. Additionally, we will explore the development of alternative personalization mechanisms in ALIMOS and effect a comparative evaluation of the advantages and disadvantages of each approach. This and other related work is currently in progress and will be reported shortly.

References

[1] W.-J. Yoon, S.-H. Oh, and K.-S. Park, "Robust music information retrieval on mobile network based on multi-feature clustering." presented at the The Second International Conference on Advanced Data Mining and Applications (ADMA 2006), ser. Lecture Notes in Computer Science, X. Li, O. R. Zai"ane, and Z. Li, Eds., vol. 4093. Springer, 2006, pp. 279–283.

[2] F. Koch and L. Sonenberg, "Using multimedia content in intelligent mobile services." presented at the WebMedia/LA-WEB 2004. IEEE Computer Society, 2004, pp. 41–43.

[3] K. Rao, Z. Bojkovic, and D. Milovanovic, *Introduction to Multimedia Communications: Applications, Middleware, Networking*. Wiley-Interscience, 2006.

[4] I. Jorstad, D. van Thanh, and S. Dustdar, "The personalization of mobile services," presented at the Proc. IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), vol. 4, 2005, pp. 59– 65.

[5] S. Y. Ho and S. H. Kwok, "The attraction of personalized service for users in mobile commerce: an empirical study," *SIGecom Exch.*, vol. 3, no. 4, pp. 10–18, 2003.

[6] S. Kang and S. Park, "A novel method providing multimedia contents according to preference clones in mobile environment." in *International Conference on Computational Science (3)*, ser. Lecture Notes in Computer Science, Y. Shi, G. D. van Albada, J. Dongarra, and P. M. A. Sloot, Eds., vol. 4489. Springer, 2007, pp. 836–843.

[7] S. Boll, J. Krosche, and A. Scherp, "Personalized mobile multimedia meets location-based services," in *GI Jahrestagung (2)*, 2004, pp. 64–69.

[8] D. Zhang and S. Shijagurumayum, "Personalized content delivery to mobile devices," presented at the Proc. IEEE International Conference on Systems, Man and Cybernetics, 2003., vol. 3, 2003, pp. 2533–2538.

[9] P. S. Lampropoulou, A. S. Lampropoulos, and G. A. Tsihrintzis, "Alimos: A middleware system for accessing digital music libraries in mobile services," presented at the Proc. KES2006 10th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems., Bournemouth , UK, October 2006.

[10] P. S. Lampropoulou, A. S. Lampropoulos, and G. A. Tsihrintzis, "Evaluation of a middleware system for accessing digital music libraries in mobile services," presented at the 5th International Conference on Information Technology : New Generations., Las Vegas, Nevada, USA, April 2008.

[11] S. Boll and J. Krosche, "mobilemm4u - framework support for dynamic personalized multimedia content on mobile systems," presented at the Proc. Techniques and Applications for Mobile Commerce (TaMoCO) Track der Multi Conference Business Informatics 2004, vol. 2769, 2004, pp. 412–421.

[12] Y.-F. Chen, H. Huang, R. Jana, T. Jim, M. A. Hiltunen, S. John, S. Jora, R. Muthumanickam, and B. Wei, "imobile ee - an enterprise mobile service platform." *Wireless Networks*, vol. 9, no. 4, pp. 283–297, 2003.

[13] A. S. Lampropoulos and G. A. Tsihrintzis, "Semantically meaningful music retrieval with content-based features and fuzzy clustering," presented at the Proc. 5^{th}International Workshop on Image Analysis for Multimedia Interactive Services., Lisboa, Portugal, April 2004.

[14] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang, "A unified framework for semantics and feature based relevance feedback in image retrieval systems," in *Proc. ACM MULTIMEDIA*, Los Angeles, California, 2000, pp. 31–38.

[15] A. S. Lampropoulos, D. N. Sotiropoulos, and G. A. Tsihrintzis, "Artificial immune system-based music piece similarity measures and database organization," presented at the Proc. 5^{th} EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Smolenice, The Slovak Republic, July 2005.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-201

Using Bibliographic Knowledge for Ranking in Scientific Publication Databases

Martin VESELY[1,2], Martin RAJMAN[1], Jean-Yves LE MEUR[2]

[1]*Ecole Polytechnique Fédérale de Lausanne*
[2]*CERN*

Abstract: Document ranking for scientific publications involves a variety of specialized resources (e.g. author or citation indexes) that are usually difficult to use within standard general purpose search engines that usually operate on large-scale *heterogeneous* document collections for which the required specialized resources are not always available for *all* the documents present in the collections. Integrating such resources into *specialized* information retrieval engines is therefore important to cope with community-specific user expectations that strongly influence the perception of relevance within the considered community. In this perspective, this paper extends the notion of ranking with various methods exploiting different types of bibliographic knowledge that represent a crucial resource for measuring the relevance of scientific publications. In our work, we experimentally evaluated the adequacy of two such ranking methods (one based on freshness, i.e. the publication date, and the other on a novel index, the *download*-Hirsch index, based on download frequencies) for information retrieval from the CERN scientific publication database in the domain of particle physics. Our experiments show that (i) the considered specialized ranking methods indeed represent promising candidates for extending the base line ranking (relying on the download frequency), as they both lead to fairly small search result overlaps; and (ii) that extending the base line ranking with the specialized ranking method based on freshness significantly improves the quality of the retrieval: 16.2% of relative increase for the Mean Reciprocal Rank (resp. 5.1% of relative increase for the Success@10, i.e. the estimated probability of finding at least one relevant document among the top ten retrieved) when a local rank sum is used for aggregation. We plan to further validate the presented results by carrying out additional experiments with the specialized ranking method based on the download-Hirsch index to further improve the performance of our aggregative approach.

Keywords: rank aggregation models, information retrieval systems, bibliometrics, scientific publication databases

Introduction

Bibliographic information has been traditionally used to evaluate and rank scientific journals. In this paper we focus on how bibliographic information and bibliometric measures can be used for ranking scientific documents within a specialized community-oriented search engine. We focus on identifying which bibliographic information and bibliometric measures are relevant for document ranking and how to integrate these measures in a more general IR model. In our work we build upon the following aggregation principle: additional specialized resources are useful for integration into an aggregated ranking model (i.e. have good possibility to improve the overall retrieval performance) if the individual ranking methods associated with the

available specialized resources do not correlate with each other, and, at the same time, individually lead to good enough retrieval results.

Within this general framework we opted for bibliometric measures that refer to documents rather than to journals (as it is the case for the journal impact factor based on citations [9] or usage [5]), and we chose the download frequency as a measure of document usage and used the associated ranking method as our baseline general purpose ranking measure. Our goal was then to identify additional ranking measures, either based on resources available for all documents, or inherent to the specialized database we used (i.e. specialized measures) that represent promising candidates for extending our base line ranking method.

In addition, we introduce a new measure - the *download-Hirsch index* - a specialized measure referring to previous authors' production based on the number of document downloads, instead on the number of citations. We experimentally analyze the suitability of freshness and of the download-Hirsch index for ranking documents in scientific publication databases and provide an experimental evaluation of the download-Hirsch index itself.

For our analyses we worked with the CERN Document Server, a scientific publication database in the domain of particle physics, covering main scientific works in the field, including published articles, preprints, conference papers, theses, scientific notes and reports, as well as other forms of scientific communication.

The rest of the article is organized as follows: in section 1 we give a brief overview of what we call specialized search engines; in section 2 we discuss the selection of bibliographic resources that are potentially interesting for ranking of scientific publications; in section 3 we present our model for extending ranking methods; in section 4 we present the experimental setup used for our tests; in section 5 we present the results of our experiments and we conclude in section 6.

1. Specialized Search Engines

The information needs of the members of a scientific community are often more specific than the ones of the general public. Indeed, document search differs when accessing particular document sub-collections that are of interest for a specific scientific community. For this reason, generic relevance criteria, such as word similarity or download frequency, do not necessarily provide optimal rankings for specialized search targeting community-specific information needs. For such purposes generic relevance criteria need to be aggregated with specialized criteria that well characterize the sub-collections in question.

Our approach is schematically depicted in Figure 1 that illustrates how a general purpose search engine operating on a document collection and using a baseline ranking method relying on resources available for all the documents (e.g download frequency) can be extended into specialized search engines operating on a sub-collection and using specialized ranking method relying on specific resources only available for the documents contained in the selected sub-collection (e.g. publication date).

For our experiments we used the CERN Document Server (CDS) that provides specialized search services for the particle physics community. Installed in the 90's CDS now comprises a comprehensive collection of documents in particle physics, covering a wide variety of document types. In order to cope with the various

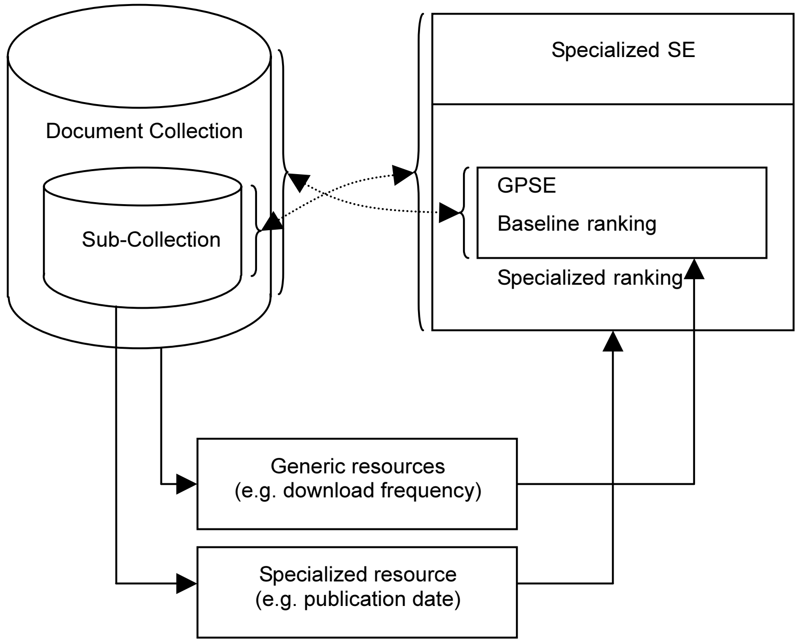


Figure1. Specialized search engine

specificities of the domain, CDS is using an in-house developed system, CDS Invenio, with many specialized indexing features and search and ranking capabilities. The current production system operates on a collection of about a million documents in particle physics, and the logged search history is available over several years in the past.

In other words, the specialized CDS Invenio search engine and the associated community-specific CDS document collection represent very well suited candidates for experiments on the extension of general purpose retrieval techniques with specialized ranking methods relying on resources derived from community specific sub-collections.

2. Selection of various ranking methods

In our work in the framework of the d-Rank project [11] we focus on the extension of generic ranking methods, used in general purpose search engines (GPSE), with specialized ranking methods based on bibliographic information only available for given sub-collections of documents. More precisely we investigate various scenarios for document rank aggregation on normalized local and global relevance scores. The ultimate objective of our project is to automatically learn aggregated ranking functions from available user access logs. In the following sub-sections we provide a brief overview of the various ranking methods considered in our project as possible candidates for aggregation. The used aggregation mechanisms are described in the next section.

2.1 General purpose ranking methods

Download frequency. The use of download data (e.g. the download frequencies) has been often considered for ranking within any GPSEs [3, 4, 5]. The idea behind the use of such type of information is the following: if document has been recently frequently downloaded, it is probable that it will remain of interest in the near future and should therefore be ranked higher in near future search results.

Word similarity. Lexical similarity between documents and queries is one of the most traditional measures of relevance used for document ranking in information retrieval. Many variations of word similarity computations have been proposed, the most advanced being based on the seminal works of Robertson and Spärck-Jones on the TF.IDF paradigm [10]. In this paradigm, TF.IDF-like measures such as the simple Cosine or the more sophisticated Okapi BM25 [16] are used to identify the most relevant documents. In our approach we can use any of the proposed similarity scores provided that the similarities between documents and indexing terms they contain can be pre-computed from the collection index.

2.2 Specialized ranking methods

Freshness. Freshness based ranking methods use temporal variables such as the *publication date* to rank scientific documents. The idea is that more recent documents are potentially more interesting for retrieval if novelty is crucial for relevance within the considered community. For example, we have discussed this issue with information specialists in the domain of particle physics and found out that, in this field, researchers indeed often tend to favor freshness as a ranking method for their information retrieval needs.

Citation frequency. Measures based on citation data are traditionally used for the ranking of scientific journals [9]. For document ranking, a simple possibility is to directly use some available journal-based measures at the document level (e.g. to use the journal score to rank the documents they contain). However, this approach raises two important issues: first journal scores (e.g. the journal impact factor) might not be discriminative enough for document retrieval; and, second, they can only be computed for documents that were published in journals. For these reasons, document *citation frequency* is often considered as a more promising resource for rankings based on citation data.

The Hirsch Index. The Hirsch index [2] based on the number of citations of individual's past production has recently been proposed as a possible measure of the impact of individual's scientific output (see formula 2.1). Similarly, as for the above mentioned journal impact factor, the standard Hirsch index might be directly used at the document level, e.g. by associating with a document the maximal Hirsch index of its authors.

$$sH(a) = \max\{h \mid \exists h \quad papers \quad p \quad s.t.(a \in A(p)) and (c(p) \geq h)\} \quad (2.1)$$

where *sH(a)* is a Hirsch index of an author *a*, *A(p)* is a set of authors contributing to the paper *p*, and *c(p)* is the citation frequency of *p*.

In the domain of particle physics, the Hirsch index for individuals and research groups has been computed by SLAC for the SPIRES database [13].

Download Hirsch Index. We define a novel Hirsch-like index, the download-Hirsch index, that relies on the same principle as the original Hirsch index with the difference that the underlying measure used for the computation is the document download frequency instead of the citation count (see formula 2.2). The idea is to compute a measure that reflects recent usage of authors' scientific output and that allows to include documents that were not published in journals, such as scientific notes, reports or theses, by computing *dH(a)* as:

$$dH(a) = \max\{h \mid \exists h \quad papers \quad p \quad s.t.(a \in A(p)) and (d(p) \geq h)\} \tag{2.2}$$

where *dH(a)* is the download Hirsch index of an author *a*, *A(p)* is the set of authors contributing to the paper *p*, *d(p)* is the number of downloads for paper *p*.

To bring the computed author-related download Hirsch index at the level of the document, we again decided, for our preliminary experiments, to use the maximum of the d-Hirsch indexes of the authors contributing to the document (see formula 2.3):

$$dH(p) = \max\{dH(a) \mid a \in A(p)\}. \tag{2.3}$$

3. Extending a ranking method

In our approach we consider that ranking methods for specialized search engines can be built as combinations of generic ranking method with more specialized ones, typically using resources that are specific to the targeted community. In this perspective we consider that a ranking method can extend (i.e. can be aggregated with) another if (1) the two methods are complementary, and (2) if the considered methods individually lead to good enough performance w.r.t. some baseline. To quantitatively evaluate complementarity, we use the Overlap@10 measure, i.e. the average size of the intersection between the top-10 documents retrieved by each of the methods for a given sample of queries. The underlying idea is that a ranking method is a good candidate for extension if its Overlap@10 is small enough to indicate that, on average, it provides enough top-10 documents that are different from the ones retrieved by the method to be extended.

Furthermore, to quantify the performance of a given ranking method, we compute two standard evaluation metrics: the Mean Reciprocal Rank and the Success@10 measure (see section 4.2). We then use the computed individual performance scores to decide whether a ranking method is good enough to be considered as an interesting candidate for extension, and to estimate the importance (i.e. the weight) the extending method should have in the extension mechanism. The extension process then proceeds in two steps: first the rank scores of individual ranking methods are normalized, and second, they are aggregated in a single ranking.

Rank score normalization. To normalize the observed rank scores, we first estimate the associated score probability density function. To do so, we use the standard technique

consisting in convoluting the observed empirical score probability distribution (i.e. the relative frequencies $f_S(s) = n_S(s)/N$, where $n_S(s)$ is the observed number of occurrences of score *s* in the available score sample *S*, and $N = |S|$ is the size the sample *S*) with a probability density kernel (i.e. any integrable function p(u) s.t. $\int_{-\infty}^{\infty} p(u)du = 1$):

$f_S * p(s) = \int_{-\infty}^{\infty} f_S(u)(p)(s-u)du = \sum_{s_i in S} f_S s_i p(s - s_i)$ and we then use the associated cumulative distribution function $cdf_{S,p}(s) = \int_{-\infty}^{s} (f_S * p)(u)du$, which corresponds to an estimate of the probability $P(S \le s)$ that the observed score *S* has a value smaller or equal to *s*, as the normalized value for the score *s*.

Rank score aggregation. As far as the aggregation of the normalized ranking methods is concerned, several types of aggregation might be considered. If scores are of homogeneous nature, an additive aggregation (such as a mixture of experts) can be used. In this case, the individual methods are aggregated in the form of a weighted arithmetic average, and the computation of appropriate weights in the average is then one of the central issues for the proper tuning of the system. For this purpose, we plan to use various regression techniques exploiting the available relevance data.

In the work presented in this paper, we investigated two possible ways of extending the ranking methods: a weighted arithmetic average of global rank scores (wGR), i.e. rank scores derived from the whole document collection; and the (equi-weighted) arithmetic average of the local rank scores (LR), i.e. rank scores derived form the subset of documents considered as presumably relevant to the processed query after an initial word similarity filtering. Notice that this local rank score method is inspired from Borda's work on vote aggregation [7], well known in Social Choice theory.

For both (local and global) approaches, the considered set of documents (i.e. the whole document collection or the output of the first step word similarity filtering) first ordered w.r.t. the considered ranking method. Document scores are then derived from the document ranks as follows:

$$s(d) = \frac{|D| - R(d) + 1}{|D|} \quad (3.1)$$

where D is the total number of documents in the considered set of documents, and R(d) is the rank of document *d* w.r.t. the considered ranking method.

In our experiments, we have extended the generic download frequency based ranking method with the freshness based one, and, for the wGR aggregation approach, we used respective weight values of {0.15, 0.85} that were chosen based on preliminary experiments evaluating the performance of various weight combinations for the aggregation. We are currently working on various learning procedures to automatically compute the required weights, the idea being that the weight of each ranking method should directly depend on the performance of this particular method with respect to some reliable performance evaluation. In our framework, performance

evaluation uses the automatically extracted referential that is described in the next section.

Finally, the table 1 below provides an overview of various, individual and aggregated, ranking methods we considered for our experiments.

Table 1. Selected ranking methods

Method	Type	Coverage
(D) Download frequency	Atomic (base line)	General purpose
(F) Freshness	Atomic	Specialized
(H) Download Hirsch	Atomic	Specialized
$(D+_{wGR}F)$	Aggregated (weighted Global Ranks)	General purpose extended to specialized
$(D+_{LR}F)$	Aggregated (equi-weighted Local Ranks)	General purpose extended to specialized
$((D+_{wGR}F)+_{wGR}H)$	Aggregated (weighted Global Ranks)	Specialized extended to specialized
$((D+_{wGR}F)+_{LR}H)$	Aggregated (equi-weighted Local Ranks)	Specialized extended to specialized

4. Experimental Setup

4.1 Experimental Data and the CDS Referential

In order to test the selected document ranking methods we used the CERN Document Server database [12] of scientific documents in the domain of particle physics. This document database covers publications amounting to a total of over one million of documents. However, for our tests, we only focused on the public collections of articles, preprints, theses, reports and scientific notes that represent about 80% of the database.

As far as evaluation is concerned, we produced a set of relevance judgments (i.e. associations between queries and corresponding relevant documents). To do so, we used the information stored in the user access logs. More precisely, we built rules operating on the sequences of user actions subsequent to the submission of a query. For example, if a document was downloaded during a search, we assumed that, with high probability, it was relevant w.r.t. the corresponding query. The resulting resource, hereafter called the CDS referential, and corresponding to a set of (query-relevant document) pairs thus allowed us to carry out performance measurements based on past user interactions with the documentary system.

It is important to notice that the CDS referential will naturally continue to grow over time, as users continue to submit new queries. We are therefore currently working on an automated extraction procedure able to periodically update the CDS referential from the growing user access logs, as well as on automated learning procedures able to compute the weights to use for aggregating the individual ranking methods with the wGR aggregation mechanism (see Fig.2 below). Applying such procedures should not

only lead to more exhaustive relevance judgments, but also, and more importantly, to more adaptive weights, allowing us to target the design of specialized search engines automatically adapting to changes in user behaviors. Similar approaches, such as Ranking SVM that uses SVM (Support Vector Machine) techniques integrating click-through data, have already been click-through data experimented, for example in [14].

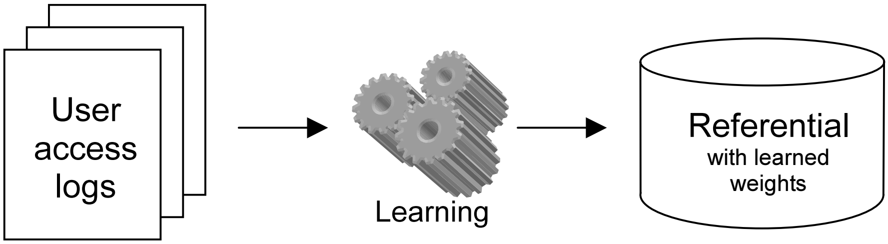


Figure 2. Learning weights from the user access logs

4.2 Evaluation metrics

When selecting the appropriate evaluation metrics, we opted for the *Mean Reciprocal Rank* and the *Success@10* evaluation measures previously used in several TREC evaluation tracks, mainly the QA and Web retrieval track [8]. For any given set of queries for which relevance information is available, the Mean Reciprocal Rank (MRR) corresponds to the average of the reciprocal of the rank of the best ranked relevant document retrieved for each of the queries, while the *Success@K* corresponds to the fraction of the considered queries for which there is at least one relevant document in the *K* best ranked retrieved documents.

Notice that the *Success@10* measure is quite well adapted to our experiments, as our search engine only outputs a default of *10* documents in its first result page and a majority of users does not navigate to next pages of the search result, only accessing documents proposed on the first result page.

Notice also that *Success@K* is different from another frequently used performance metrics, the *Precision@K*, which corresponds to the average fraction of relevant documents found on the top-K result list. One of the important reasons for us not to use Precision/Recall based metric is that our current evaluation resource – the CDS referential – often contains incomplete relevance judgments which might strongly bias Precision and Recall results.

5. Preliminary results

In our experiments, we first evaluated the complementarity of the selected ranking methods by computing *Overlap@10* values for various ranking method pairs. The table 2 below presents the average *Overlap@10* values obtained for the selected ranking method pairs for a set of test queries extracted from the CDS referential.

Table 2. Overlap@10 for various ranking method pairs

Ranking method 1	Ranking method 2	Overlap@10
(D)	(F)	20%
(D)	(H)	19%
(F)	(H)	16%
(H)	(D+F)	16%

As all observed overlap values appeared to be quite low (between 16% and 20%), we considered that all the selected methods were potentially interesting candidates for extension, and we therefore carried out individual performance evaluations (in terms of the *Success@10* and *MRR*) for each of them (see table 3 below).

Table 3. Performance of individual ranking methods

	Generic	Success@10	MRR
(D)	Download frequency	0.532	**0.229**
	Specialized	**Success@10**	**MRR**
(F)	Freshness	**0.593**	0.173
(H)	d-Hirsch	0.365	0.138

As these evaluations indicated that freshness (F) leads to a better performance than the download-Hirsch index (H), we decided to first extend the base-line ranking method (D) with freshness (F), and, in a second step, to further extend the best performing aggregated method (the (D+F) method using Local Rank aggregation, as it can be seen in table 5) with the download Hirsch index (H).

As shown in the table 4 below, we found out that, for both evaluation metrics, the best performing method was the (D+$_{wGR}$F) method, i.e. the download frequency combined with freshness through the Local Rank aggregation mechanism, which achieved an MRR of 0.266 and a Success@10 of 0.623. It is interesting to notice that the Local Rank aggregation mechanism (LR) was in general superior to the weighted Global Rank aggregation mechanism (wGR).

Table 4. Performance of aggregated ranking methods

Ranking method	Aggregation type	*Success*@10	MRR
(D+F)	wGR	0.599	0.226
	LR	**0.623**	**0.266**
((D+$_{wGR}$F)+H)	wGR	0.452	0.184
	LR	0.533	0.223

However, as the performance results observed for individual and aggregated methods often appear to be very close (for example, the (D) and the $((D+_{wGR}F)+_{LR}H)$ methods for both *MRR* and *Success@10*), it was therefore necessary to further validate the obtained results with a statistical test measuring whether the observed performance differences were indeed statistically significant. For this, we carried out a 10-fold cross validation of the performance results in order to perform a paired t-test (after having checked the required normality and equal variance assumptions).

The tables 5 and 6 below present the performance results obtained for the 10 generated test samples, while the Fig.3 and Fig.4 present the results of the significance tests (with the significance levels indicated as percentages on the arrows) for both of the selected evaluation metrics.

Table 5. 10-fold cross-validation for Success@10

	D	**F**	**H**	**$(D+_{wGR}F)$**	**$((D+_{wGR}F)+_{wGR}H)$**	**$(D+_{LR}F)$**	**$((D+_{wGR}F)+_{LR}H)$**
1	0.530	0.595	0.383	0.591	0.451	0.635	0.532
2	0.500	0.610	0.375	0.629	0.477	0.617	0.496
3	0.534	0.583	0.320	0.583	0.409	0.636	0.561
4	0.568	0.618	0.390	0.618	0.467	0.637	0.564
5	0.500	0.545	0.379	0.572	0.436	0.576	0.500
6	0.504	0.582	0.363	0.590	0.473	0.600	0.533
7	0.532	0.577	0.341	0.566	0.434	0.603	0.487
8	0.543	0.597	0.353	0.605	0.438	0.612	0.508
9	0.553	0.625	0.360	0.617	0.458	0.664	0.577
10	0.556	0.595	0.386	0.625	0.475	0.653	0.568
Mean	**0.532**	**0.593**	**0.365**	**0.599**	**0.452**	**0.623**	**0.533**
StD	**0.023**	**0.022**	**0.021**	**0.021**	**0.021**	**0.025**	**0.032**

Table 6. 10-fold cross-validation for MRR[1]

	D	**F***	**H***	**$(D+_{wGR}F)$**	**$((D+_{wGR}F)+_{wGR}H)$***	**$(D+_{LR}F)$**	**$((D+_{wGR}F)+_{LR}H)$**
1	0.224	0.169	0.145	0.204	0.178	0.254	0.218
2	0.227	0.177	0.125	0.235	0.191	0.276	0.229
3	0.248	0.171	0.135	0.223	0.175	0.293	0.264
4	0.238	0.167	0.134	0.218	0.181	0.281	0.230
5	0.215	0.164	0.147	0.216	0.178	0.246	0.218
6	0.214	0.170	0.124	0.225	0.174	0.239	0.216
7	0.240	0.179	0.149	0.242	0.205	0.256	0.224
8	0.219	0.170	0.121	0.231	0.183	0.239	0.187
9	0.225	0.181	0.147	0.230	0.188	0.287	0.216
10	0.244	0.187	0.150	0.237	0.185	0.287	0.235
Mean	**0.229**	**0.174**	**0.138**	**0.226**	**0.184**	**0.266**	**0.224**
StD	**0.012**	**0.007**	**0.011**	**0.011**	**0.008**	**0.020**	**0.018**

[1] Columns marked with a star (*) correspond to performance result sets that did not pass the normality test that is, in theory, a necessary pre-requisite for applying the selected statistical test.

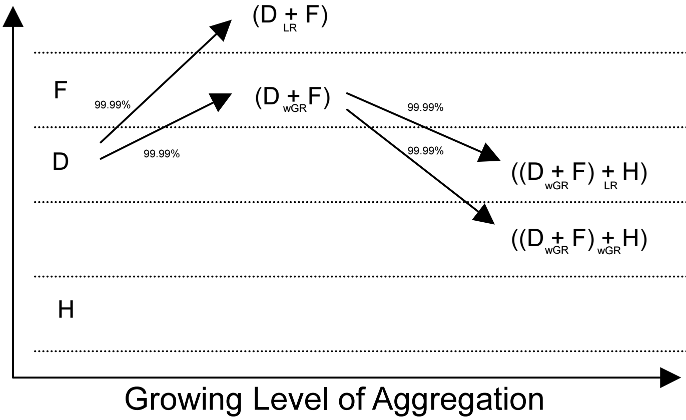


Figure 3. Results of the paired t-test for the Success@10 performance results

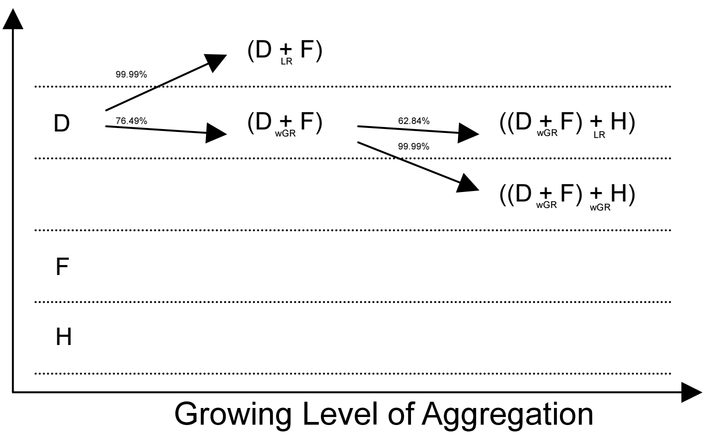


Figure 4. Results of the paired t-test for the MRR performance results

The carried out statistical significance tests confirmed that the ranking method combining download frequency and freshness with the Local Rank aggregation mechanism performs significantly better than the base line ranking, for both the *MRR* and the *Success@10* performance measures. For several other aggregated ranking methods no significant difference w.r.t. the base line was observed. For example, the (D$+_{\mathrm{wGR}}$F$+_{\mathrm{LR}}$H) method (resp. the (D$+_{\mathrm{wGR}}$F) method) performs at a comparable level to the base line for both performance measures (resp. the *MRR* performance measure), suggesting that for these methods aggregation currently does not provide any significant improvement w.r.t. the base line.

6. Conclusions and future work

In our experiments we have shown that, as measured by *Overlap@10* values, freshness and the download-Hirsch index provide useful complementary information to the download frequency base line ranking method used in the CDS search engine. In terms of retrieval performance measured on the CDS referential, the MRR scores

obtained by the individual ranking methods were of ~0.226 for the download frequency ranking method, ~0.174 for the freshness and ~0.138 for the download-Hirsch ranking method. In addition, when extending the download frequency by freshness we obtained a superior MRR performance of ~0.266.

The CDS referential used for our experiments has been automatically extracted from user access logs, and we plan to proceed with the implementation of an automated learning procedure using the CDS referential to achieve more precise and possibly better results through the optimized computation of the weights used for the Global Rank aggregation mechanism. In addition we also plan to carry out tests using a standard IR collection with complete relevance judgments.

As far as other future work is concerned, we plan to further investigate the download-Hirsch index, in particular by computing such an index for groups of co-authors instead of individual authors. We indeed believe that such an approach should lead to more discriminative index values, and, in this perspective, we plan to compare the performance of the corresponding specialized (atomic and aggregated) ranking methods with both the results presented in this paper, and the ones obtained with the standard citation based Hirsch index, as well as with other bibliometric indexes.

Finally, another important aspect concerns the fact that, so far for our index computations, we have only taken into consideration downloads done from the CERN Document Server, and not the ones corresponding to documents, available on the CERN Document Server, but accessed through some other connected server, such that the web site of a publisher or another preprint server in the domain of particle physics such as ArXiv.org or SPIRES. In order to compute more representative overall usage indexes, we thus envision implementing some variation of a distributed or federated approach for usage-based ranking.

References

[1] Jacobs R.A., Jordan, M.I., Nowlan S.J. and Hinton G.E.: Adaptive mixtures of local experts. *Neural Computation*, **3** (1991), 79–87.

[2] Hirsch J: An Index to Quantify an Individual's Scientific Research Output. *PNAS* **102** (2005), 16569-16572.

[3] Darmoni S.J., Roussel F., Benichou J., Thirion B., and Pinhas N.: Reading factor: a new bibliometric criterion for managing digital libraries. *Journal of Medical Library Association,* **90** (2002), 323-327.

[4] Bollen J and Van de Sompel H.: Usage Impact Factor: the effects of sample characteristics on usage-based metrics. *Journal of the American Society for Information Science and technology,* **59** (2006), 001-014.

[5] Bollen J, Rodriguez A, Van de Sompel H.: MESUR: user-based metrics for scholarly impact. *JCDL'07,* 2007.

[6] Lotka A. J.: The frequency distribution of scientific productivity. *Journal of the Washington Academy of Sciences* **16** (1926), 317–324.

[7] Borda J.C. : Mémoire sur les élections au scrutin, Histoire de L'Académie Royale des Sciences. 1784.

[8] Craswell N and Hawking D. Overview of the TREC-2004 Web Track. *Proceedings of the 13th Text Retrieval Conference*. 2004.

[9] Garfield E: Citation Analysis as a tool in journal evaluation. *Science* **178** (1972), 471-479.

[10] The IDF page, http://www.soi.city.ac.uk/~ser/idf.html

[11] d-Rank web site, http://liawww.epfl.ch/d-rank/

[12] CERN Document Server, scientific publication database, http://cdsweb.cern.ch/

[13] SPIRES, scientific publication database, http://www.slac.stanford.edu/spires/hep/

[14] Joachims T: Optimizing Search Engines Using Clickthrough Data. *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD),* ACM, 2002.

[15] Silverman B W: Density Estimation for Statistics and Data Analysis. CRC Press, 1986.

[16] Robertson S: Okapi at TREC-3. *Proc of the TREC-3 conference*. 1994.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-213

The roles of tool and human mediators in Web-based course of self-regulation

Raija LATVA-KARJANMAA
University of Helsinki

Abstract: *The study focused on identifying the roles of tool and human mediator on a web-based course of developing better learning skills, i.e. self-regulative skills. Mediators act as supporters of the learning process. The assumption is that in virtual environments mediators can have particular roles, which are different from a face-to-face learning situation. Online tool mediators can support the student's learning skills, which has been proved earlier and again in this study. In addition to this the mediators can build awareness of possibilities to develop as learners, mediate a feeling of personal space, freedom and integrity. and also act as a mediators of guidance, support and control.*

Keywords: Mediators, mediation, self-regulation, online tutoring, learning to learn

Introduction

The aim of this study is to find out, what kind of particular mediators are experienced by students in asynchronous online study course on learning skills i.e. self-regulation. A self-regulated learner is aware of her cognitive processes, goal-oriented, able to change her learning strategy according to her needs and able to evaluate her own study success. In Zimmerman's words "self-regulated learning (SRL) refers to the self-directive processes and self-beliefs that enable learners to transform (develop and improve) their mental abilities [1]. Rheinberg, Vollmeyer and Rollet [2] remark that the *search for mediators* in different learning situations and learning tasks is an important next step when studying self-regulation. Mediators are human [3], social, symbolic or tool [4] supporters of learning, which assist the learner in the learning process. It is likely that a net-based environment differs from a face-to-face learning environment when related to the learning helpers, the mediators. While an increasing amount of higher educational studies are completed in technology-based learning environments, mediators can have a particular role in the learning process. Net-based environments call for students who are independent and able to master their studies with less tutor control i.e students with good self-regulative skills. These skills can be critical to academic success in online courses according to Williams and Hellman [5]. Also the faculty needs to have an idea which students will exercise self-regulation skills effectively [ibid].

That self-regulation has an impact on *academic* success on the whole has been demonstrated continually [6]. Promoting learning with various *tools online* has been studied during the last decade. Winne et al. [7], developed a course environment with gStudy, which enables learners to make notes, create glossaries, label and index content, construct concept maps, search for information, chat and collaborate, and receive coaching. The potential of the gStudy for assisting students to use self-regulation

strategies is impressive according to Zimmerman [1] and Nenniger, who emphasizes its usability in any subject [8]. Azevedo et al.[9] developed an sophisticated think-aloud method for assessing students' self-regulated learning processes as an online event in a hypermedia learning environment (HLE), which according to Zimmerman [1] have much potential for improving learning. He remarks nevertheless, that HLEs require personal initiative and technical skills. Both the gStudy tool and think-aloud method are good mediators of self-regulative skills but they are *labour intensive constructs to the academic course designer and students*. HLEs require students who are skilful users of technological tools and also students whoa are already quite self-directed. The IQ FORM project of Niemi et al. [10] designed the interactive Web-based tool '*The IQ Learn*' to support learners' self-regulation in virtual universities and other higher education settings. IQ Learn is an interactive tutoring tool to support learning online. The motivation for the development of this environment has been to offer a relatively simple learning platform, which can be used by any student or tutor in virtual universities. The IQ FORM services are offered in Finnish, English and Swedish. In the IQ FORM Learn pilot, Niemi, Nevgi and Virtanen [11] observed that students in higher education vary in their self-regulation abilities and receive little or no tutoring for their learning skills. The pilot use of the IQ Learn found that students benefited from virtual tutoring. A net based tool of self-regulation can obviously enhance learning, especially at the beginning of academic studies. This study focuses on the IQ Learn environment [12]. The IQ FORM self-assessment tools are based on research and the development of EDUFORM, which is an intelligent questionnaire software based on Bayesian modelling [13] and adapted as the technological platform of the IQ FORM. The IQ Learn tool consists of three elements (1) IQ FORM Test tool, with three questionnaire sets for students' self-evaluation, (2) IQ FORM Tutoring tool, with a hypertext structure tutoring students towards self-regulation and additional guidelines for teachers and also (3) IQ FORM diary tool for the reflection of learners' experiences and test profiles.

1. The theoretical background: Mediators in the self-regulation process

Rheinberg, Vollmeyer and Rollet have been *searching for mediators* in the self-regulation process [14] and claim that they have a motivational role in the initial (goal) phase of the study process. Below the concept of mediation is viewed. The concept of mediation has important role in semiotics, especially in writings of Peirce [15] and his presentation of semiosis enlightens the mediation as the basic process of human learning in a trans-scientific way. The concept of mediation also has strong connotations with role of *new media in learning*. The media philosophical critique of Heidegger [16] and also lately Baudrillard [17] on information technologies as distorting the human reality are well known. Recently *media ethical* issues have been addressed by Introna [18].The current media philosophical discussion of Margrieter and media pedagogical writings of Hug discuss the "*medializierung*" [19] and the *mediatic turn* [20] of the human culture. Sandbotne's media philosophical writings [21] concern the transformational role of the media as a tool for changing human thinking processes.

In *learning psychology* many researchers do not theorize about the difference between the concept of *mediation*, the concept of *scaffolding* introduced by Bruner [22]

or even *tutoring* (especially used in web-based learning support) in psychology of learning. Mediation is the basic construction of learning according to pioneering research of mediation in the context of learning psychology; Feuerstein [3],Vygotsky [4]. Their ideas have been developed further by Kozulin [23], Wertsch [24] and Säljö [25]. All teaching and learning involves mediation according to Feuerstein [3] and Vygotsky [4]. Based on Vygotsky and Feuerstein, Kozulin and Pressesein [26] define that a mediator selects, changes, amplifies, and interprets objects and processes of the human learning. Vygotsky [4] brought the idea of *tool, symbolic and social* mediators to the analysis of the learning process. Feuerstein [3] concentrated mostly in human mediators. Kozulin [23] has advanced these ideas by calling the tool and human mediators for the agents of mediation and calls for the analysis of the qualities of those mediators. The entire learning environment can be seen as a tool mediator and the Net Tutor as a human mediator.

2. The research question

As could be seen in research of Winne and Azevedo (above) mediators in form of advanced course tools can improve the learnings skills of as student. It has also been shown in the pilot study of the IQ LEARN in Finnish Virtual University the IQ Learn is a tool, which increases the self-regulation of a student. The interest in this study is however to analyse what kind of *particular roles* the mediators can have. Can the mediators be described more in detail as having particular roles in net-based learning. 1. *What is the role of Tool mediator* (IQ Learn) in learning self-regulation 2. *What is the role of Human mediator* (Net Tutor, usually called tutor) in learning self-regulation.

3. The research setting, research methods and data collection

The empirical research setting includes a research intervention of 15 students from Northern Carelia University in Finland. The students were studying in a Web-based course of learning to learn with IQ FORM tools during four weeks in spring 2003. The study environment was IQ Learn of the IQ FORM system and the researcher acted as a Net Tutor. A semi-structured interview of the students was conducted after they completed the IQ FORM course. The method used in this study was design research [27] and its adaptation to educational research by De Corte and Verschaffel [28]. The data was collected with semi-structured interviews. The data analysis was realized the qualitative phenomenological data analysis method of Moustakas [29] and its further developed by Cresswell and Moerer-Uhrdahl [30].

4. The findings

The analysis of 450 statements of the students concerning their experiences during the course gave as a result that *the particular roles of the human and tool mediators during the online course were*

i) to act as mediators an awareness of the student's existing study skills of possibilities to develop as a learner. *Tool and Human mediator*
ii) to act as mediators of personal space, freedom and integrity. *Tool mediator*
iii) to act as mediators development of learning skills. *Tool mediator*
iv) to act as mediators of guidance, support and control. *Human mediator*

i) Both the tool mediator (IQ Learn) and the human mediator (Net Tutor) added to the *awareness new possibilities* of the course. Majority of the students became aware of new kinds of learning opportunities in three perspectives: (1) course goals, (2) their existing study skills and the gaps in these skills (3) possibilities to develop as learners with the net based course. The net *IQ Learn as tool mediator* was solely in charge of adding the awareness of own existing study skills. It was experienced as useful while it revealed the existing strengths and weaknesses of the students' study skills trough the IQ Learn Tests and showing them how they could train their learning skills trough the IQ Learn Tutoring sets. The *human mediator* was better when explaining the goals of the net course, while many of the students said that they experienced some difficulties in understanding the course goals just based on the information in the net.

ii) IQ FORM course environment (the tool mediator) created a *personal working and thinking space*. The students experienced that the Internet learning environment gave them the possibility to chooce the time and place for studies, which added to their sense of *freedom*. The students could explore her/ his personal thinking in peace without being influence by or observed and possibly critiqued by a tutor or peer students, which created a feeling of personal *integrity*.

iii) The IQ FORM tutoring sets gave the students a lot of hints, practical examples and exercises on how to develop their learning skills. Eight (8) students said that they gained better exam preparing skills, six (6) students became better in expressing their thoughts in writing, seven (7) students said they learned to control their test anxiety better, five (5) students improved their time management and five (5) students developed their skills as critical thinkers.

iv) The students needed "a face", a person to connect the course with. The surprising information gained from the interviews of the students was a feeling of trust and integrity with a Net Tutor, when compared to a regular classroom tutor. Many students experienced that they dared to write more personal things in their learning diaries about their studies and their thoughts about themselves. They felt a kind of anonymity but also a personal contact. The students said that the Net Tutor was not experienced as a teacher. The Net Tutor was as an important support and guide but also a controller, although she was not needed as a teacher.

References

[1] Zimmerman, B.J. (2008) Self-regulation on-line measures in *American Educational Research Journal* Vol.45, No. 1, pp. 166 –183. AERA: New York

[2] Rheinberg, F., Vollmeyer, R. & Rollett, W. (2000). Motivation and action in self-regulatedlearning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation*. San Diego, CA: Academic Press.

[3] Feuerstein, R. & Feuerstein, S. (1991). Mediated learning experience: A theoretical review. In Feuerstein, R., Klein, P. & Tannenbaum, A. (Eds.). *Mediated learning experience*: Theoretical, Psychological, and learning implications. Tel Aviv and London: Freund

[4] Vygotsky, L. (1978). Mind in Society. Harvard university press: Cambridge.

[6] Zimmerman, B., Schunk D. (1989) (Eds) Self-regulated Learning and Academic Achievement: Theory, research, and practice. Springer: New York. Hofer, R, B., Y, S. L. & Pintrich, P. (1998) Teaching

college students to be self-regulated learners, in: D. H. Schunk & B. J. Zimmerman (Eds) *Self-regulated* Learning.
[7] Winne, P. H. (2006). How software technologies can improve research on learning and bolster school reform. *Educational Psychologist*, 41, 5-17.
[8] Nenniger P (2006) Comment on Perry & Winne's "Learning from Learning Kits: gStudy Traces of Students' Self-Regulated Engagements with Computerized Content". *Educational Psychological Review* 18: 233-237.
[9] Azevedo, R., Cromley, J. G., & Siebert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia? *Contemporary Educational Psychology*, 29, 344–370.
[10] Niemi, H. & Ruohotie, P. (Eds.), Theoretical Understandings for Learning in the Virtual University. Research Centre for Vocational Education, University of Tampere. Saarijärvi.
[11] Niemi, H., Nevgi, A., Virtanen, P. (2003) Towards Self-Regulation in Web-Based Learning. Journal of Educational Media, Vol. 28, No. 1, March 2003
[12] www.edu.helsinki.fi/iqform/default_eng.htm and
[13] Nokelainen, P., & Tirri, H. (2002). Issues in Designing an Interactive Personalized Self-Assessment Tool in H. Niemi & P. Ruohotie (Eds.), Theoretical Understandings for Learning in the Virtual University. Research Centre for Vocational Education, University of Tampere. Saarijärvi.
[14] Vollmeyer, R. & Rheinberg, F. (2006). Motivational effects on self-regulated learning with different tasks. Educational Psychology Review. 18, 239-253.
[15] The Peirce Edition Project (Eds.). (1998). The Essential Peirce: Selected Philosophical Writings, Volume 2. Bloominton & Indianapolis: Indiana University Press.
[16] Heidegger, M. (1977). The question concerning technology. In Krell, D. F. (Ed.), Martin Heidegger: Basic Writings, 283-317. San Francisco: HarperCollins Publishers
[17] Baudrillard, The Vital Illusion (2000). New York: Columbia University Press.
[18] Introna L, (2005) Phenomenological approaches to ethics and information technology. In The Stanford Encyclopaedia of Philosophy. First published Sat 19 Feb, 2005
[19] Margreiter, R. (1999). Realität und Medialität - zur Philosophie des Medial Turn In: Medien Journal 1/ 1999:9-18.
[20] Hug, T. (2005b). Pisa und Medienkompetenz. PISA und andere Erinnerungen. Beiträge der Medienpädagogik zur Vermeidung von „educational lags". Ein Plädoyer für zukunftsorientierte Reformen unter den Auspizien des „mediatic turn". Medienimpulse, Heft nr 53.
[21] Sandbothe, Enquiries at the Interface: Philosophical Problems of Online Education. A Special Issue of The Journal of Philosophy of Education, edited by Paul Standish und Nigel Blake, Oxford, Blackwell, 2000, pp. 53-69.
[22] Bruner, J. S. (1975). From communication to language: A psychological perspective. Cognition, 3, 255–287.
[23] Kozulin, A. (1998). Psychological tools a sociocultural approach to education. Cambridge: Harvard University Press. Kozulin, A., Gindis, B., Agejev, V., Miller, M. (Eds.). (2003). Vygotsky's Educational Theory in Cultural Context. Cambridge: Cambridge University Press.
[24] Wertsch, J. (1995). Sociocultural Studies of Mind (Learning in Doing: Social, Cognitive & Computational Perspectives). Cambridge: Cambridge University Press.
[25] Säljö, R. (1995). Mental and Physical Artifacts in Cognitive Practices. In Reiman, P., Spada (2005). *Learning in Humans and Machines: Towards an Interdisciplinary Learning Science*. Oxford: Pergamon/Elsevier
[26] Kozulin, A. & Pressesein, B. (1995) Mediated Learning Experience and Psychological Tools: Vygotsky's and Feuerstein's Perspectives in a Study of Student Learning in Educational psychologist 1995, 30(2), 67-75. Lawrence Erlbaum Associates, Inc.
[27] Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in educational research. Educational Researcher, 32(1), 9-13.
[28] De Corte, E. (2003). Designing learning environments that foster the productive use of acquired knowledge and skills in De Corte, E., Verschaffel, L., Entwistle & Van Merrieboer, J. (Eds.) (2003). *Unravelling Basic Components and Dimensions of Powerful Learning Environments*. Elsevier.
[29] Moustakas, C. (1994). Phenomenological Research methods. Thousand Oaks: Sage
[30] Moerer-Urdahl, T., & Creswell, J. (2004). Using transcendental phenomenology to explore the "ripple effect" in a leadership mentoring program. *International Journal of Qualitative Methods*. International Journal of Qualitative Methods, 3 (2). Article 2

Development of Multi-Modal Interfaces

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-221

Human Identification and Positioning System Based on the Combination of Motion Sensor and Video Camera

Junichi Tajima [a], Kimio Shintani [b], Hirohide Haga [a] and Shigeo Kaneda [a]
[a] *Gradual School of Engineering, Doshisha University*
[b] *Department of Early Childhood Education, Tokiwakai College*

Abstract. In this paper, we propose a method of sensor data processing that detects the direction and the stride of walking. We have proposed a positioning system using only a combination of passive motion sensors and image data, but the precision of the positioning of the previous method was insufficient. The previous method, which counted the number of steps from the up-and-down motion of the foot, had severe limitations. It only could detect forward motion, and therefore could handle less flexibility of walking. In order to solve this problem, we paid attention to the movement of the swing phase and the stance phase of walking and detected the direction and the stride of walking, using data of horizontal acceleration. The proposed method recognizes movement to the front and rear, and to the right and left. Acquired sensor data will be collated with the image data from a camera. Based on the prototype system, the experimental evaluation of the proposed method is demonstrated.

Keywords. positioning, image processing, sensor, activity tracking, infant education, ubiquitous computing

1. Introduction

Ubiquitous application systems based on position information are currently under investigation. The mainstream approach in existing systems includes detecting a person's position by using electromagnetic waves such as Active Badge or RFID tags. There is also a growing field of applying information about a person's indoor position. One such application would be to infant education in nursery schools or childcare centers.

However, there is an issue regarding the application of ubiquitous technology for infant education—the consent of parents. It has been proven that weak electromagnetic waves will cause no health problems. However, parents and schoolteachers often do not accept the usage of electromagnetic appliances for emotional reasons.

For such reasons, we started to develop a positioning system with no emission of electromagnetic waves. We have used a combination of motion sensors and video

cameras, which are completely passive, to detect users' locations, and proposed a method of identifying and positioning using the data from these devices [1].

However, the precision of the positioning of the sensor data was insufficient. It was a method that counts the number of steps from the up-and-down motion of the foot. The issue was not being able to recognize a direction except the forward moving. The sensor was attached to the waist.

In this paper, we propose a new method of sensor processing that detects the direction and the step of walking, using the data of horizontal acceleration. The sensor is attached to the ankle. This method makes free walking possible and the precision of the positioning of the sensor data is improved.

To evaluate the proposed method of combinations of sensor and camera, we have built a prototype. The evaluation result indicates that highly accurate information on human identification and positioning is acquired using the method.

2. Previous Method

2.1. Evaluation of Related Works

As for related works, there are GPS, RFID technology and the like. There is a system that visualizes the communication and activity situation in an organization using an acceleration sensor and an infrared sensor [2]. Moreover, research into a system that specifies the location of children by RFID tags is established [3].

For adjustment to childcare facilities, we use radio-free devices. For conducting analysis of children, accuracy within 10 to 30 cm is required in positioning. For those reasons, the use of motion sensors and video cameras is promising.

2.2. Overview of the Positioning System Based on the Combination of Motion Sensor and Video Camera

Table 1 shows each advantage and fault of sensor data and image processing. It is found that independent, highly precise positioning and private identification can't be achieved. We aim to create a passive, highly accurate and identifiable positioning system, combining sensor data analysis with image processing.

Figure 1 shows a schematic diagram of our proposed system. Our proposed method operates according to the following steps.

1. Collect 3-dimensional acceleration data from a motion sensor. These data include a sensor ID, and we apply these data to identification in STEP 3.
2. Take the position data from camera images. These data have accurate position data, but it is not possible to identify the moving object.
3. Integrate the two sets of information, thus giving identifiable positioning.

Table 1. Feature of a Sensor Data and Image Processing

Method	Accuracy	Personal Identity
Sensor Data	Low	Possible
Image Processing	High	Impossible

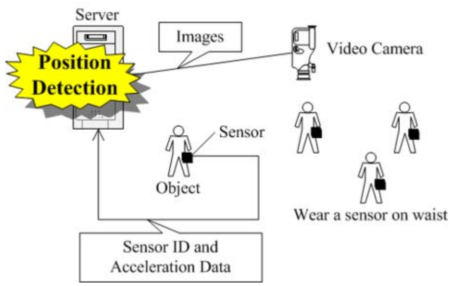


Figure 1. Image of Our Proposed System

3. Issues with Our Previous Method

With the system using a motion sensor and video camera proposed before, the precision of the positioning of the sensor data was insufficient. It was a method that counts the number of steps from up-and-down actions, adding a severe limitation to of subjects only being able to move forward in a constant stride.

3.1. Sensor Data Processing of Previous System

The axes of the acceleration sensor swing with the object's action. To calibrate the axes with images taken using a fixed camera, the system corrects the axes using the rotation data from the sensor's initial location. The sensor's coordinates are inputted to the Euler angle (Z-Y-X coordinates). The system translates Euler coordinates into camera coordinates. It then rotates the acceleration vectors along camera axes, and integrates the acceleration data to positioning data.

The acceleration integration method comes under the influence of a drifting offset. The errors accumulate in an experiment. To solve this problem, we have adopted the pedometer method to count the number of steps resulting from up-and-down actions such as walking. In addition, the system detects an object's position by direction of movement and moving distance per unit of time. The sensor is attached to the waist. The system operates according to the following steps.

1. This sensor outputs the A/D data of nine axes (three acceleration axes, three angular velocity axes, and three terrestrial magnetism axes). The unit of acceleration is converted to gravity acceleration, the angular velocity to deg/s and the terrestrial magnetism to a quantity without unit to treat data easily.
2. The acceleration in the perpendicular direction (direction of the Z-axis) is stable at minus 1 while people are standing when an upward position is assumed to be a plus. When walking it changes to become biggest at the time the foot lands. Therefore, when the acceleration exceeds the threshold, it is thought of as one step. The stride assumes 0.7 meters of the definite value. Applying the step and the number of steps, it presumes the advanced distance.
3. The progress direction is estimated with the terrestrial magnetism of the X-axis and Y-axis. The angle from true north is calculated.

3.2. Issues of Sensor Data Processing of Previous System

The previous system performs incorrect position tracking. When a horizontal walk or a backward walk is carried out, the system detects it as a pre-walk. In addition, the pace is assumed to be constant. There was no versatility, and high-precision sensing of the position was difficult.

4. Proposed Sensor Data Processing

The improved sensor data processing is proposed in this chapter. It is an improved pedometer method that recognizes movement front and rear, right and left, and achieves highly precise positioning.

4.1. Outline of Proposed Sensor Data Processing

The proposed sensor data processing pays attention to the swing phase and the stance phase of walking. The motion sensor attached to the waist previously is attached to the ankle. Figure 2 shows the axial direction of the acceleration of the sensor attached to the foot.

Acceleration date is acquired from the attached sensor and the processing, one-step detection, direction detection and step detection are performed.

One-step detection is the processing to cut the portions assumed to be one step from the data one-by-one. Direction detection is the processing to determine the isolated data of the person moving in any direction. And step detection is the processing to detect a step. Next, this processing is described in detail.

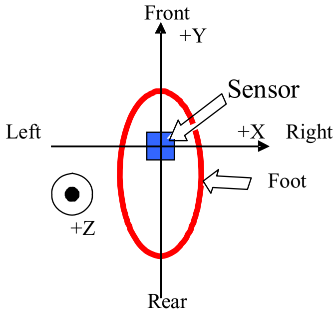


Figure 2. Axial Direction of Sensor

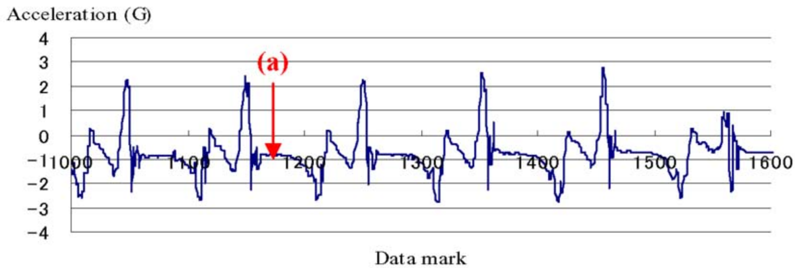


Figure 3. Z-axis Acceleration at Time of a Pre-walk

4.2. One-step Detection

In order to extract the portions assumed to be one step from the obtained time series data, it is necessary to detect the landing of a leg. The data of the acceleration in the perpendicular direction (Z-axis) is used.

Figure 3 is Z-axis acceleration data at the time of a pre-walk. When the 20 or more states where the difference of the absolute value of two continuous data is 0.1 or less continuously (Figure 3 (a)), it is judged as landing.

However, the same phenomenon is seen, even if the leg has not landed when gradually walking. In this case, since the generating interval of a phenomenon is very short, therefore a short interval is disregarded. And the data during landing is taken and it is assumed one step.

4.3. Direction Detection

The direction is detected toward which the leg with a sensor stepped. It processes by dividing in the case of two.

The first is the case where it changes a lot before and after the direction in which the toe tip has turned to step forward. In this case, the method of our previous method is adopted because the precision of the method relying on geomagnetism in detecting the direction is good.

When the direction of the toe tip is the same before and after stepping forward, the following processing is performed on the data obtained.

1. The data of the X-axis and Y-axis acceleration in the section assumed to be one step is extracted.
2. The moving average in 20 points is calculated from the data respectively.
3. The absolute value of the difference between the maximum value of the moving average and the minimum value (the amplitude difference) is calculated.

Table 2 shows the standard that judges the direction of a walk. The reason for adding judgment of motion to the front and rear, and to the right and left is to distinguish stepping.

The direction decision method of a slanting walk is described. It calculates direction with Eq. (1) of the X-axial direction as 0rad. However, it is calculable only from 0rad to 90rad by Eq. (1).

Then, the feature of the time serial accelerating data is utilized. When becoming the maximum value after the minimum value in the data of the X-axis and the Y-axis, it is considered as a pattern-1. Let the contrary be a pattern-2. The area stepped forward is decided using the combination of this pattern of X-axis and Y-axis data (Figure 4). If the area is understood, it is possible to calculate by applying each of Eqs. (1), (2), (3), and (4) to area-1, 2, 3, and 4.

$$\arctan(\text{amplitude difference of Y-axis} / \text{amplitude difference of X-axis}) \tag{1}$$

$$\arctan(-\text{amplitude difference of Y-axis} / \text{amplitude difference of X-axis}) \tag{2}$$

$$\arctan(\text{amplitude difference of Y-axis} / -\text{amplitude difference of X-axis}) + \pi \tag{3}$$

$$\arctan(\text{amplitude difference of Y-axis} / \text{amplitude difference of X-axis}) - \pi \tag{4}$$

Table 2. Walk Judging Standard

Walk of Front and Rear	Total of amplitude difference of X-axis acceleration and Y-axis acceleration is 1.1G or more and difference is 0.5G or more.	Pre-walking	Y-axis acceleration drop and rise.
		Back-walking	Y-axis acceleration rise and drop.
Walk of Front and Rear		Right walking	X-axis acceleration rise and drop.
		Left walking	X-axis acceleration drop and rise.
Slanting Walk	Both X-axis acceleration and Y-axis acceleration is 0.85G or more, or the total is 1.1G or more and the difference is 0.5G or less.		

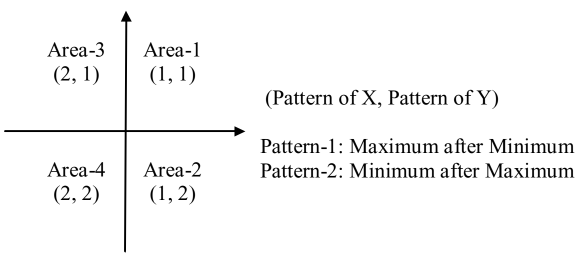


Figure 4. Area Stepped Forward

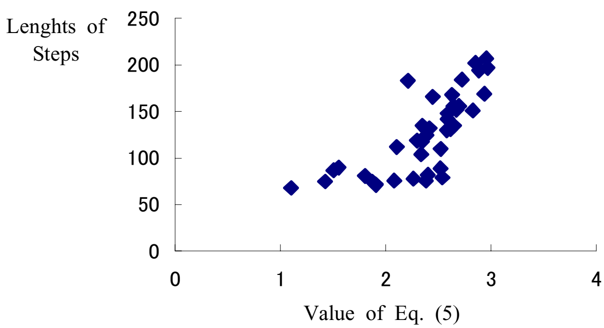


Figure 5. Relation between Measurement Value and Value of Eq. (5)

4.4. Detection of the Stride

The judgment of the stepping is described. When the total of the amplitude difference of X-axis acceleration and Y-axis acceleration is 1.1G or less, it is judged as a standstill and the stride is set to 0.

Equation (5) is used for detection of the stride. It turned out that this equation is suitable from 40 experiments. As a result of the experiment, the relation between the lengths of steps and the value of Eq. (5) is shown in Figure 5. From this figure, it can be expected that the correlation is among two values. The correlation coefficient was actually set to 0.8, that is, there was high correlation. Then a regression line is calculated from this data. It is used for detection of the stride.

$$\sqrt{\text{amplitude difference of X - axis}^2 + \text{amplitude difference of Y - axis}^2} \qquad (5)$$

5. Image Data Processing

Image data processing is performed using the difference between frames as a comparatively easy technique. We implement the following processes:

1. Obtaining the difference between frames
2. Binarization, smoothing, and interference rejection
3. Labeling the consecutive area
4. Calculation of the coordinates of the center of gravity for each label
5. Integration of the neighboring area

Objects with coordinates near the center of gravity between frames are considered the same object. In that case, some problems exist. One is the problem that it will not be extracted if an object stands still. If the object does not exist in the frame, it is judged that the object is standing still. The coordinate of the front frame is left. And pursuit becomes possible even if an object stands still.

Another is a problem in case two or more objects exist to be close in the next frame. Then, it supposes that those objects may be the same objects, and a course is branched. The pattern of a course of objects can be covered by this processing. However, since the number of courses increases very much by this processing, processing time is divided into 20 seconds respectively.

6. Collation

The collation method of sensor processing data and video processing data is described. At first, based on the coordinates of a route, all of the correlation coefficients of the routes obtained by each processing is derived. Next, the combination of a correlation coefficient that becomes the largest is chosen. The selection determines the route of the person with the attached sensor obtained by video processing.

In addition, the search of a route of video processing is a 20-second interval. The course of a long time is acquired by connecting the routes in which the correlation coefficient is the highest.

7. Evaluation

The evaluation experiment was conducted using the improved pedometer method we proposed.

7.1. Experimental Conditions

Three subjects (students) participated in the experiments, and one of them had a sensor unit by NEC-TOKIN [4] connected to a laptop PC. The PC recorded sensor data with time stamps. We took images with an overhead camera from a height of 7m. The field of activity was limited to the range (6x5m) of the camera, and the subjects moved freely within that range. Each experiment lasted 40 sec.

7.2. Experimental Result

The experiment for 40 seconds was divided into 20-second parts. Let each be section-1 and section-2. The result collated in each section is shown in Table 3.The highest correlation coefficient value in each section is shown. The route that unites section-1 and section-2 and shows the highest correlation is extracted. This route must be the correct route of the person with a sensor. It is the solution obtained by our proposed method.

It is compared with the value of the sensor and the actual route, which shows a correct value made by visual observation. Figure 6 shows the result of each X and Y coordinate. Furthermore the average error of the coordinates acquired in the proposed method and the correct coordinates was calculated (Table 4).

Table 3. Highest Correlation Coefficient Value in Each Section

	Section-1 (0-20sec)	**Section-2 (21-40sec)**
X	0.93	0.78
Y	0.81	0.64
Average	0.87	0.71

Table 4. Average Error of Coordinates

	X-coordinate	**Y-coordinate**	**X-Y plane**
Section-1	11.43	16.08	24.28
Section-2	15.78	16.57	22.88
Total	13.61	16.16	23.58

7.3. Consideration of Experimental Result

The possibility of being a route of a person with sensor is high if correlation with the route of a sensor is high. Table 2 shows that there is high correlation in each section. It turns out that the extracted route and the correct route are almost equal from Figure 6. It shows that although the route of a sensor has a large error, the route extracted by the proposed method has a small error. Table 4 shows an error average is about 24 cm, and the validity of this method is confirmed.

7.4. Future Assignment

In the above implementation, the motion sensor was connected to a note PC, but this was too large for an infant to carry around. Therefore, we must develop a sensor unit that is durable, has enough data capacity for one day's activity, and a small enough size that it does not limit the wearer's activity.

In this work, we set one camera above the activity area, but in a real field, it would not always be possible to set the camera directly above. Additionally, the activity area would not be as limited as it was in the kindergarten. Moreover, only one camera causes limitations in practical use. Consequently, in future work we plan to use multiple cameras. Future work will involve using stereo cameras for shooting in the near-horizontal plane, and, moreover, we are eyeing the possibility of using a far-infrared camera for moving-object detection. This type of camera ignores background noise such as outside light.

Moreover it is necessary to define the processing of the penetrating and leaving to a photography area. The judgment of a frame-out and a frame-in is performed in collation with sensor data.

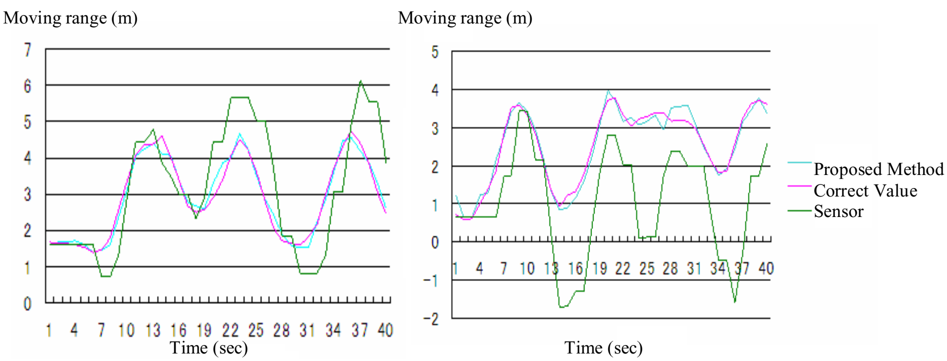


Figure 6. Comparison of X (left) and Y (right) Coordinates

8. Conclusions

We focused on positioning using motion sensor and video camera. This method can accurately detect each person's position and identity by collating sensor data and image processing. And there is an advantage that position specification can be performed without putting out electromagnetic waves.

The sensor attached to the waist previously was attached to the ankle. The direction and the stride are detected from X-axis and Y-axis acceleration time series data of the sensor. Positioning corresponding to people's flexible walking patterns is achieved, and the accuracy of the position is improved.

Image data processing is performed using the difference between frames. From the position information on the extracted person it searched for the route as much as possible, and the continuous position information of the person is acquired.

As the result of having used the correlation coefficient and having performed positioning of an individual based on these routes, the error average was about 24cm. Therefore, the evaluation result indicates that highly accurate information of human identification and positioning is acquired using the method.

References

[1] Jun Kawai, Kimio Shintani, Hirohide Haga and Shigeo Kaneda, Identification and positioning based on Motion Sensors and a Video Camera, the 4th IASTED International Conference on WEB-BASED EDUCATION, No. 461-809, February 21-23, 2005

[2] Co. Hitachi. Ltd, Business Kenbikyou, http://www.hitachi.co.jp/New/cnews/month/2007/06/0622.html, 2007

[3] Y Takasugi, Recording children's activities using a radio tag system, Workshop by Kyushu Private Kindergarten Federation

[4] NEC-TOKIN, http://www.nec-tokin.com/

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-231

How efficient is it to communicate customer knowledge with the aid of audio-visual metaphors?

Mutlaq ALOTAIBI, Dimitrios RIGAS
m.b.g.alotaibi@bradford.ac.uk, d.rigas@bradford.ac.uk
Department of computing, school of informatics, University of Bradford, UK

Abstract. The shift in customer role from a source of revenue to a source of innovation has resulted in an increased demand for a direct interaction with customers, via more advanced and real-time solutions. This may drive organisations to complement Electronic Customer Knowledge Management Systems (E-CKMS) with new interactive multimodal metaphors. There is little known about CKM empirical studies, which investigated whether multimodal metaphors as part of a CKM interface can outperform the text with graphics only metaphors. Thus, this paper describes an investigation about the role of multimodal interaction in improving E-CKMS efficiency compared to text with graphics only metaphors. The examination of this hypothesis was carried out empirically on two E-CKMS experimental platforms (text with graphics only and multimodal) designed for this study. Two independent groups of users (n=20 for each group) evaluated the two interface versions of E-CKMS. Results suggested that the multimodal CKM interface was more efficient than the text with graphics only interface.

Keywords. Customer, Knowledge, Multimodal, Interaction, Efficiency, Sound, Speech, Earcons, Auditory Icons, Audio-visual

Introduction

Today's business environment has witnessed increasing levels of organisations complexity and volatility of markets due to the exponential growth of knowledge [1]. Although, the volume of knowledge triples every eighteen months, its concept is still not completely well-defined [1]. Knowledge can be elicited from sources within the organisational boundary and beyond [2], and the latter is the most valuable one according to Osterle [3]. That is to say, market studies prepared by departments of research and development (RnD) are regarded as less beneficial for the organisation compared with the manner in which the utilisation of CK cuts costs and improves performance [4]. Gurgul et al [5] listed the high cost of RnD as well as the shorter product lifecycle among the driving forces of CK exploitation. When managers became aware of the importance of CK (listen to the customer's voice), they encountered the flow of CK into the organisation from several communication channels (email, fax, and phone), and hence the integration and management of multiple communication channels was proposed [6]. This proposal, in fact, led to the synergy of a technology-enabled management tool with a social-oriented method labelled as Customer Relationship Management (CRM), and Knowledge Management (KM) respectively in E-Business contexts [6, 7]. The determination and characteristics of CRM, KM and E-Business are regarded as beyond the scope of this paper, and can be found in [1, 7-9].

The term E-CKMS is widely used to refer to the integration between KM and CRM in E-Business contexts [7]. Many authors, such as Gibbert et al [4], Gurgul et al [5] and Rollins and Halinen [10], introduced Amazon case study as a typical example of E-CKMS, and uncovered several CRM and KM aspects. Consider for example, Community of Customers (CoC) that guides the customer to make accurate purchase decisions based on expertises of peer customers in a form of reviews and ratings [4]. This concept is derived from Community of Practice (CoP), which is introduced in the context of traditional KM [5, 11]. Another example related to CRM is that Amazon supports the knowledge-based interaction with E-Business users, in which the powerful analytical abilities of CRM systems analyse customers historical data to leverage up-selling and cross-selling opportunities [7, 12]. Therefore, studies that aim to improve the performance of E-CKMS can consider Amazon as a benchmark.

Interactive systems are among the CRM components that facilitate E-CKM, and can contribute to the expansion of CK exploitation, because CK is better to be elicited in the absence of human (sales representatives) mediation that leads to the misinterpretation of customer needs. Gibbert et al [4] supported this view, and argued that conveying what customers really want can not be achieved by relying on the understanding of sales representatives or the results obtained from market research that involved high cost and conducted less frequently. Massey et al [13] argued that customer interaction should be established in a form of two-way dialogue to facilitate what so-called learn while interacting. In addition, Lesser et al [11] stated that not only the establishment of two-way dialogue, but also the creation of CoC can widen the use of CK. In summary, E-CKMS that incorporate means of interactive technologies have the potential to perform better than traditional E-CKMS.

The multidisciplinary nature of E-CKMS has raised a number of practical concerns derived the original disciplines (CRM and KM). Examples of these are the identification of relevant knowledge, knowledge hoarding and trust, and information overload. Several suggestions and solutions were proposed to either alleviate or tackle these issues, such as looking for characteristics of knowledgeable customer [14], customer rewards systems [4], building loyalty schemes [15], Customer Innovation Toolkits (CIT) [16, 17], and implementing interactive systems [2, 4-6, 18]. According to Gibbert et al [4] interactive multimedia systems were proposed to address trust and interactivity questions. Furthermore, information overload and document relevancy can be regarded as problems associated with the web-based environment. In order to tackle information overload [2, 6], interface designers can associate sounds with information represented visually allowing the system to have continuous real-time sound support [18].

The remainder of the paper is organised in six sections. In Section 1, we introduce relevant work. Section 2 describes the experimental platform. Design of the empirical study is shown in Section 3. In Section 4, we present results and discussion. Conclusion is provided in Section 5. Finally, we describe future work in Section 6.

1. Relevant work

Previous research carried out in the field of CKM focused on theoretical frameworks and models, which can be organised into three main categories: CK-based, Business Engineering (BE) [3] and CKM-based models. In CK-based models, CK characteristics, approaches, applications and relations was identified and described [8, 11, 19, 20]. In

BE-based models, authors described four separate levels, which are strategy, process, system and change levels, and focused on the concept of process-orientation [6, 21, 22]. In CKM models, Gibbert et al [4] introduced the five styles of CKM, while Feng and Tian [20] focused on types of CK. García-Murillo and Annabi [23] believe that CK is best interpreted by humans, and proposed a CKM process. In summary, previous CKM studies did not pay enough attention to the empirical evaluation of the role of multimodal interaction in E-CKMS. The remainder of this section describes research that aimed to evaluate the exploitation of multimodal interaction in several fields that are relatively close to E-CKMS.

There is little known about efforts devoted to evaluate the efficiency of multimodal interactive E-CKMS. Nevertheless, the significance of this approach was found in many other relatively related fields [24-28]. In the field of software engineering, Sonnenwald et al [29], Cohen and Ludwig [30], DiGiano et al [31], Rigas et al [32], and Rigas and Alty [33] were advocates for this view. Typically, multimodal metaphors have the potential to improve the performance of general Information Systems (IS). Burke et al [34] conducted a meta-analysis research to study the role of multimodal interaction by gathering data from forty three studies, and found that this manner of user interaction outperform the visual-only one. Additionally, findings obtained from studies conducted by Rigas and Memery [35] supported this result, by examining the use of multimodal interaction metaphors (speech, earcons and auditory icons) to communicate output of general IS [36]. Rigas and Memery [35] concluded that participants not only were able to interpret a mixture of speech and sound successfully, when communicated concurrently, but also could cope with a higher volume of visually presented information reinforced with auditory metaphors. In addition, the utilisation of rising pith metaphors to convey graphical information was investigated in two studies [26, 37], and found advantageous for the interpretation of visually-impaired users to graphical information in the absence of visual metaphors.

The work done in web-based browsing and email environments can be regarded as relatively relevant to E-CKMS field due to the similarity it involves. In web-based browsing systems, Fernström and McNamara [38] developed a musical tones browsing prototype aided by sound to examine the role of sound in browsing, and found that sound-aided manner of browsing significantly improved the system efficiency. In addition, an experimental speech-enabled online help system was designed and tested in order to introduce guidelines for designing user interface for such kind of systems [24], and was extended afterward by incorporating further non-speech and other auditory metaphors [39]. In email data browsing, several experiments [35, 40, 41] were undertook to evaluate the utilisation of multimodal interaction in reducing email visual complexity, communicating email categories, and other email-related information. The findings revealed that visual complexity was reduced and usability was improved by communicating information aurally. In addition, in order to tackle additional concerns of information hiding and the overwhelming of visual channel, Rigas [41] investigated the utilisation of audio-visual metaphors, and found that the visual complexity of email data browsing can be reduced by means of the synchronisation of visual and audio representation.

Table 1. E-CKMS visual and auditory metaphors

CK Category	VCKMS		MCKMS					
	T	G	T	G	S	E	A	R
Trends (top 10)		√		√		√		
Customer reviews	√		√		√		√	
Customer ratings		√		√		√		
Website advice		√		√		√	√	
Co-production CK	√		√	√	√	√	√	
Product features	√	√	√	√	√	√		√

2. Experimental platform

This study involved developing an experimental E-CKMS, which provides typical functions often found in web-based mobile phones retailing, and was implemented with two interfaces. These were text with graphics only E-CKMS (VCKMS) and Multimodal E-CKMS (MCKMS). Typically, E-CKMS consist of back-end, infrastructural and front-end (E-CKMS Interface) components, and might introduce a function labelled as *co-production* [4], which can be best defined as the manner of New Product Development (NPD) in which the customer acts as both consumer and producer at the same time. Co-production incorporates solution space, innovation toolkits and trail-and error engine, and facilitates not only NPD, but also real-time and interactive elicitation of CK. Co-production solution space can be composed of several elements (parameters of tariff scheme), manipulated by innovation toolkits in order to produce trails (billing schemes), which afterward can sent to trail-and-error engine (billing engine). This cycle represents the process of NPD that were fully implemented, and the reason behind choosing this kind of products is because of the electronic nature that enables E-CKMS to produce trails in the absence of complete line of production. Due to the scope and aims of this research, focus was put upon the system interface, and an assumption was made that back-end, infrastructural components were fully implemented.

The E-CKMS Interface was designed and implemented differently based on the two environments: VCKMS and MCKMS. There were several types of CK communicated to E-CKMS users (visually, auditory or simultaneously), which were organised into six categories: trends, customer reviews, customer ratings, website advices, co-production CK and product features. In addition, there were two visual metaphors employed: Text (T) and Graphic (G), and four auditory ones: synthesised speech (S), earcons (E), auditory icons (A) and recoded speech (R). In order to incorporate sounds into E-CKMS, a great deal of technologies, tools and sounds has been used, such as musical notes, speech agent, text-to-speech engine, environmental sounds [42], sound recoding software [43] and multi-timber synthesiser software [44]. The empirically derived guidelines provided by Brewster [45] were followed in the creation of earcons. Families of earcons was differentiated by employing timber (*guitar, violin, trumpet, drum, organ and piano* [46]), and used to communicate different types of CK, and further differentiation was made by utilising rising pitch metaphors.

Table 2. Summary of task complexity factors

Task level	CKM Activity		Complexity factors			
	Code	**Description**	**CKI**	**CI**	**NOTR**	**NOAS**
Ease (Task E)	A1	Ease phone selection activity	Low	Low	6	18
	A2	Ease tariff selection activity	Low	Low	4	22
Moderate (Task M)	A3	Moderate phone selection activity	Moderate	Moderate	7	8
	A4	Moderate tariff selection activity	Moderate	Moderate	5	9
	A5	Co-production with two trails	Moderate	Moderate	3	N/A
Difficult Task (D)	A6	Difficult phone selection activity	High	High	7	2
	A7	Difficult tariff selection activity	High	High	4	2
	A8	Co-production with five trails	High	High	6	N/A

For example, guitar and violin were mapped to the top 10 lists CK, in order to communicate the best and worst rated products respectively, and rising pitch would convey the product position in both lists. In addition, the environmental sounds used were sound of *typing, cheering, clapping, laughing, gasping, fog horn, side whistle and camera shot*. Table 1 shows the mapping between CK types and metaphors in the two E-CKMS versions.

3. Design of empirical study

This research carried out an investigation about different task complexity levels: task easy (Task E), task moderate complex (Task M) and task difficult (Task D). Although, task difficulty is subjective in nature, five aspects were devoted to distinguish levels of complexity. These are the number of task requirements (NOTR), the number of available selections (NAOS), CKM activities (CKMA), CK intensity (CKI) and customer interaction (CI). NOTR represented how many task requirements needs to be fulfilled in order to consider the task as successfully completed, while NAOS was to refer to the number of available products that when selected by the user, the task is regarded as accomplished. When the task was designed to be difficult, NOTR was increased, while NAOS was decreased. It was categorised also based on CI and CKI into E (A1 and A2), M (A3, A4 and A5) and D (A6, A7 and A8), which represented levels of complexity. Types of CKMA were phone selection, tariff selection and Co-production. There have been eight CKMA in total, divided based on CKMA type into three phone selections (A1, A3 and A6), three tariff selections (A2, A4 and A7) and two co-production activities (A5 and A8). In A3, for example, the user is provided with this scenario (Say that your phone preferences are: The phone should be among the top10 or website advice lists. The phone should be a camera phone (capacity between 0.5 and 3MP exclusive), and a 3G phone. The number of positive reviews should be greater than the negative ones.). Burke et al [34] provided more information on task levels, types and workload. Table 2 summarises complexity factors and shows how tasks are linked to CKMA.

Fortify users (all were students at University of Bradford, and regular internet users) performed the three tasks and completed a questionnaire devised of several

knowledge-related questions. Selection of subjects has followed the non-probability sampling strategy and convenience-sampling method [47]. Subjects were offered a short training session prior to the experiment, and were divided into two independent groups of twenty each. Group C (control) performed all tasks using VCKMS, and Group E (experimental) used MCKMS to finish tasks. The order in which the three tasks were performed was balanced between users in order to alleviate the effect of task learnability. Variations of the tasks orders were coded by specific technique that indicates the E-CKMS version used and in which order tasks were performed. V-EMD, for example, performs task E, M and D in the same order using VCKMS. The two E-CKMS versions were evaluated and all users performed all tasks. Results of this examination are presented and discussed in the next section.

4. Discussion of results

Associating knowledge representation with sound can aid users to find targets faster and perform more accurate co-production, with fewer actions compared to the text with graphics only one. Subjects expressed interest in using audio-visual interface, but found it difficult when audio messages were first communicated. However, when users adapted to the structure of auditory messages, sound appeared to be useful modality for conveying CK and supporting customer decisions. Furthermore, it was noteworthy that the extent in which auditory stimuli (speech, earcons and auditory icons) facilitates efficient communication of CK involved variation. Auditory icons, for example, showed a significant contribution in aiding user performance when it was used to present CK that have well-known association with environmental sounds in real life. On the other hand, earcons appeared to be, to some extent, less useful because it could not be interpreted as naturally as auditory icons, but it was very helpful in conveying CK that have relatively low range of values, such as customer ratings. Speech, in addition, was expected to contribute significantly to support E-CKMS users to accomplish CKMA that involve reading very long texts, such as customer reviews.

Figure 1 shows the mean values of (A) percent of task accomplishment per unit of time, and (B) task accomplishment time for the three tasks with the values for using VCKMS and MCKMS. The unit of time used in order to calculate values of these two factors was one hundred seconds. In Figure 1.A, trends of task accomplishment per unit of time vary slightly in task E, whereas in task M and D, there have been a marked rise in its variance. In task E, the mean value for VCKMS was only slighter than for MCKMS (17%). In contrast, the mean values for MCKMS in tasks M and D were considerably lower than that for VCKMS. Significance of the difference between the two versions was tested using the t-test. The difference was found significant (t_{37} = 5.066, CV= 2.03, $P < 0.05$). In Figure 1.B, it can be seen that difficult tasks were accomplished significantly faster with the sound-aided manner of CK representation than text with graphics only one. In task E and M, the mean value for VCKMS is only slighter than for MCKMS (18%). In contrast, the mean value for task D shows a very different picture. The task accomplishment time for MCKMS was remarkably lower than that for VCKMS (22%). The t-test results show that there was significant difference between the two versions (t_{37} = 6.004, CV= 2.03, $P < 0.05$). Therefore, it can be said that the significant difference goes hand in hand with the task levels, and the influence of audio-visual metaphors on both task accomplishment time and the percent of task accomplishment per unit of time was remarkably significant for difficult tasks.

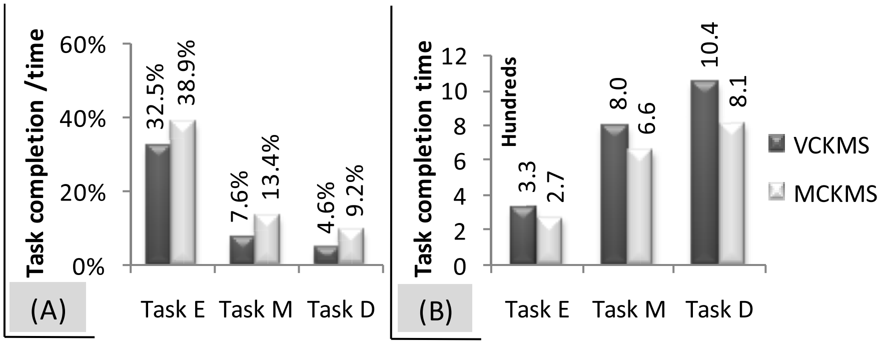


Figure 1. Mean values of (A) percent of task accomplishment per unit of time, and (B) task accomplishment time (in 100 seconds), for the three tasks using VCKMS and MCKMS

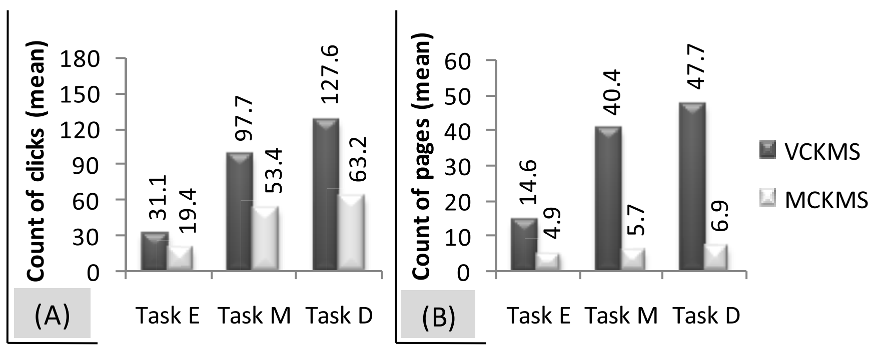


Figure 2. Mean values of number of actions ((A) count of clicks, and (B) count of visited pages) for the three tasks using VCKMS and MCKMS

Figure 2 shows the mean values of the number of actions (A- count of mouse clicks, and B- count of visited pages) required to accomplish the three tasks with the values for using VCKMS and MCKMS. In Figure 2.A, there was a marked reduction in count of clicks needed to finish tasks using MCKMS, compared with VCKMS. In task E, the value for MCKMS was significantly lower than that for VCKMS (38%). In task M and D, it can be seen that the mean values for VCKMS were just double that for MCKMS. In addition, t-test results showed a highly significant difference between the two conditions in terms of mouse clicks required to accomplish all tasks (t_{34} =17.4, CV= 2.03, $P < 0.05$). To be brief, the count of clicks in E-CKMS can be cut in half by the augment of visually displayed CK with auditory metaphors. In Figure 2.B it was noteworthy that the count of visited pages when using MCKMS fall into a narrow range in all task complexity levels, whilst, in VCKMS, the more complex the task is, the more pages users visit. In task E, the mean values for VCKMS was almost triple that for MCKMS. If the mean values for were compared in tasks M and D, it can be seen that those for VCKMS were almost sevenfold those for MCKMS. Furthermore, the t-test results showed a highly significant difference with regard to the number of pages visited in order to complete all tasks (t_{20} = 26, CV= 2.09, $P < 0.05$). Therefore, it can be said that levels of the number of pages that need to be visited in order to complete tasks using MCKMS were relatively insensitive to task difficultly levels, in comparison with VCKMS.

The occurrence of errors was also analysed for the three tasks with the values for using VCKMS and MCKMS. In general, it was noticed that the occurrence of errors in E-CKMS was reduced by incorporating multimodal metaphors, essentially in the sophisticated tasks requirements. In task E, it was observed that there was no difference in the mean value of error rate between the two versions (VCKMS= 1.1, MCKMS= 1). In task M and D, the mean values of error rate for VCKMS (task M= 5.3, task D= 9) were almost double that for MCKMS (task M= 2.8, task D= 4.8). According to t-test results, a significant difference between VCKMS and MCKMS was found with regard to errors occurrence in both task M (t_{34} =4.3, CV= 2.03, $P < 0.05$) and task D (t_{28} = 8.5, CV= 2.05, $P < 0.05$), but not in task E (t_{36} = 0.369, CV= 3.03, $P > 0.05$). In brief, the employment of audio-visual metaphors in E-CKMS reduced the occurrence of errors, especially when tasks that involve higher levels of user interaction were performed.

5. Conclusion

This research explored four efficiency-related aspects (the time taken and the number of actions required to accomplish tasks, as well as the percent of task completion per unit of time and error rate) of multimodal interaction in E-CKMS. Due to the multi-disciplinary nature of E-CKMS, extreme care must be taken in order to control the flow of CK from several communication channels. Human misinterpretation and lack of trust are two of the concerns that led to proposing direct, structured and dialog-based extraction of CK. Thus, the potential that interactive technologies have to maximum the volume of communicated CK is evaluated empirically. During the experiment phase, efficiency attributes provided insights into the significance of multimodal metaphors in improving E-CKMS user's performance. The experience with E-CKMS gained from this study suggested that the resourceful interpretation of knowledge could be facilitated by means of multimodal interaction because knowledge either is intensive, and hence not easy to be perceived visually, or communicated more rapidly. Therefore, it is important for E-CKMS interface designers to employ multimodal interaction not as full replacement of text with graphics only communication technique, but as additional complementing interaction metaphors.

6. Future work

The efficient communication of CK types via avatar, facial expressions and body gestures is one of the issues that needs further investigation. Therefore, the future experimental work is expected to suggest new ways in which facial expressions as well as body gestures can be used to convey CK, identify further research directions and provide empirically derived guidelines that aid the collection and management of CK. In addition, fewer types of CK (trends, customer reviews, website advices and product features) are expected to be identified as a basis for the investigation with the focus upon activities of product selection.

References

[1] I. Becerra-Fernandez, A. J. Gonzalez, and R. Sabherwal, Knowledge management : challenges, solutions, and technologies. Upper Saddle River, N.J.: Pearson/Prentice Hall, 2004.

[2] T. Davenport and L. Prusak, Working knowledge : how organizations manage what they know. Boston, Mass.: Harvard Business School Press, 1998.

[3] H. Osterle, Business in the Information Age: Springer, 1995.

[4] M. Gibbert, M. Leibold, and G. Probst, "Five Styles of Customer Knowledge Management, and How Smart Companies Use Them To Create Value," European Management Journal, vol. 20, pp. 459-469, 2002.

[5] G. Gurgul, M. Rumyantseva, and E. Enkel, "Customer Integration–Establish a constant bilateral knowledge flow," Diskussionspapier des Research Centers Knowledge Source der Universität St. Gallen, 2002.

[6] A. Bueren, "Improving performance of customer-processes with knowledge management Adrian Bueren, Ragnar Schierholz, Lutz M. Kolbe, Walter Brenner The Authors," Business Process Management Journal, vol. 11, pp. 573-588, 2005.

[7] A. Tiwana, The essential guide to knowledge management : e-business and CRM applications. Upper Saddle River, NJ. London: Prentice Hall PTR ; Prentice-Hall International, 2001.

[8] D. J. Skyrme, Capitalizing on knowledge : from e-business to k-business. Oxford: Butterworth-Heinemann, 2001.

[9] M. Alotaibi and R. Alzahrani, "Evaluating e-Business Adoption: Opportunities and Threats," Journal of King Saud University, vol. 17, pp. 29-41, 2004.

[10] M. Rollins and A. Halinen, "Customer Knowledge Management Competence: Towards a Theoretical Framework," System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on, pp. 240a-240a, 2005.

[11] E. Lesser, D. Mundel, and C. Wiecha, "Managing Customer Knowledge," Journal of Business Strategy, vol. 21, pp. 34-7, 2000.

[12] S. L. Pan and J. N. Lee, "Using e-CRM for a unified view of the customer," Communications of the ACM, vol. 46, pp. 95-99, 2003.

[13] A. P. Massey, M. M. Montoya-Weiss, and K. Holcom, "Re-engineering the customer relationship: leveraging knowledge assets at IBM," Decision Support Systems, vol. 32, pp. 155-170, 2001.

[14] G. L. Urban and E. von Hippel, "Lead User Analyses for the Development of New Industrial Products," Management Science, vol. 34, pp. 569-582, 1988.

[15] C. Dennis, D. Marsland, and T. Cockett, "Data mining for shopping centers—customer knowledge management framework," Journal of Knowledge Management, vol. 5, pp. 368-374, 2001.

[16] E. von Hippel, "User Toolkits for Innovation," Journal of Product Innovation Management, vol. 18, pp. 247-257, 2001.

[17] E. Von Hippel, "Learning from Open Source Software," Sloan Management Review, vol. 42, pp. 82-86, 2001.

[18] S. A. Brewster, "Using non-speech sound to overcome information overload," Displays, vol. 17, pp. 179-189, 1997.

[19] J. E. Rowley, "Reflections on customer knowledge management in e-business," Qualitative Market Research: An International Journal, vol. 5, pp. 268-280, 2002.

[20] T. X. Feng and J. X. Tian, "Customer Knowledge Management and Condition Analysis of Successful CKM Implementation," Machine Learning and Cybernetics Proceedings of 2005 International Conference on., vol. 4, pp. 2239- 2244 2005.

[21] R. Bose and V. Sugumaran, "Application of knowledge management technology in customer relationship management," Knowledge and Process Management, vol. 10, pp. 3-17, 2003.

[22] M. Dous, H. Salomann, L. Kolbe, and W. Brenner, "Knowledge Management Capabilities in CRM: Making Knowledge For, From and About Customers Work," presented at the Eleventh Americas Conference on Information Systems, Omaha, NE, USA, 2005.

[23] M. García-Murillo and H. Annabi, "Customer knowledge management," Journal of the Operational Research Society, vol. 53, pp. 875-884, 2002.

[24] A. Kehoe and I. Pitt, "Designing help topics for use with text-to-speech," Proceedings of the 24th annual conference on Design of communication, pp. 157-163, 2006.

[25] D. I. Rigas, D. Memery, D. Hopwood, and M. A. Rodrigues, "Using non-speech sound to communicate information in user interfaces," Applied Informatics 2000, pp. 357–362, 2000.

[26] D. Rigas and J. Alty, "The rising pitch metaphor: an empirical study," International Journal of Human-Computer Studies, vol. 62, pp. 1-20, 2005.

[27] W. W. Gaver, "Auditory Interfaces," Handbook of Human-Computer Interaction, vol. 1, pp. 1003-1041, 1997.

[28] M. Cohen, "Throwing, pitching and catching sound: audio windowing models and modes," International Journal of Man-Machine Studies, vol. 39, pp. 269-304, 1993.
[29] D. H. Sonnenwald, B. Gopinath, G. O. Haberman, W. M. Keese Iii, and J. S. Myers, "InfoSound: an audio aid to program comprehension," System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conference on, vol. 2, 1990.
[30] M. Cohen and L. F. Ludwig, "Multidimensional audio window management," International Journal of Man-Machine Studies, vol. 34, pp. 319-336, 1991.
[31] C. J. DiGiano, R. M. Baecker, and R. N. Owen, "LogoMedia: a sound-enhanced programming environment for monitoring program behavior," Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems, pp. 301-302, 1993.
[32] D. I. Rigas, J. L. Alty, and F. W. Long, "Can music support interfaces to complex databases," EUROMICRO-97, New Frontiers of Information Technology, pp. 78–84, 1997.
[33] D. I. Rigas and J. L. Alty, "Using sound to communicate program execution," Proceedings of the 24th EUROMICRO Conference, vol. 2, pp. 625–632, 1998.
[34] J. L. Burke, M. S. Prewett, A. A. Gray, L. Yang, F. R. B. Stilson, M. D. Coovert, L. R. Elliot, and E. Redden, "Comparing the effects of visual-auditory and visual-tactile feedback on user performance: a meta-analysis," Proceedings of the 8th international conference on Multimodal interfaces, pp. 108-117, 2006.
[35] D. I. Rigas and D. Memery, "Utilising audio-visual stimuli in interactive information systems: a two domain investigation on auditory metaphors," Information Technology: Coding and Computing, 2002. Proceedings. International Conference on, pp. 190-195, 2002.
[36] D. Rigas, H. Yu, and D. Memery, "Experiments Using Speech, Non-Speech Sound and Stereophony as Communication Metaphors in Information Systems," euromicro, 2001.
[37] J. L. Alty and D. Rigas, "Exploring the use of structured musical stimuli to communicate simple diagrams: the role of context," International Journal of Human-Computer Studies, vol. 62, pp. 21-40, 2005.
[38] M. Fernström and C. McNamara, "After direct manipulation---direct sonification," ACM Transactions on Applied Perception (TAP), vol. 2, pp. 495-499, 2005.
[39] A. Kehoe, F. Neff, and I. Pitt, "Extending traditional user assistance systems to support an auditory interface," Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications table of contents, pp. 637-642, 2007.
[40] D. I. Rigas and D. Memery, "Multimedia e-mail data browsing: the synergistic use of various forms of auditory stimuli," Information Technology: Coding and Computing [Computers and Communications], 2003. Proceedings. ITCC 2003. International Conference on, pp. 582-586, 2003.
[41] D. Rigas, "Empirically Derived Multimedia Design Guidelines for Browsing Large Volumes of E-Mail Data," 2003.
[42] W. W. Gaver, "Auditory Icons: Using Sound in Computer Interfaces," Human-Computer Interaction, vol. 2, pp. 167-177, 1986.
[43] KYDsoft, "Easy MP3 Sound Recorder," 3.1 ed, 2006.
[44] AGR, "Virtual Music," 1.0 ed, 2005.
[45] S. A. Brewster, P. C. Wright, and A. D. N. Edwards, "Experimentally derived guidelines for the creation of earcons," Adjunct Proceedings of HCI, vol. 95, pp. 155-159, 1995.
[46] D. Rigas and J. Alty, "How Can Multimedia Designers Utilise Timbre?," presented at HCI-98, People and Computers XIII, Springer, British Computer Society, Sheffield, UK, 1998.
[47] N. J. Salkind, "Sampling and generalizability," Exploring research, vol. Sixth edition, pp. 85-98, 2006.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-241

Using Graphical Representations to Improve the Usability of Email Clients

Dimitrios RIGAS, Saad ALHARBI
Department of Computing, University of Bradford, BD7 1DP, Bradford, UK
D.Rigas@Bradford.ac.uk, S.T.Alharbi@Bradford.ac.uk

ABSTRACT. With the versatility of email and the growing number of email messages, email clients have become difficult to use. Information visualization has been employed by many studies for various purposes such as the employment of the management features to email clients. Nevertheless, graphical presentation of email data has not been significantly considered to improve the usability of email clients. This paper describes an empirical study that aimed to investigate whether the usability of email clients can be improved by visualising email data. Two experimental e-mail visualisation approaches were developed especially for this experimental programme, both of them graphically presenting email messages based on a dateline together with other email data. Thirty users were required to perform ten experimental tasks in the two experimental visualisations and in a typical email client. Tasks accomplishment time and number of actions carried out whilst performing the tasks in order to evaluate the usability of each email version. The results indicated that the graphical presentation of email messages can significantly improve the usability of email clients. However, it can also negatively affect the usability if the email data was hidden extensively to reduce the graphical complexity in the inbox.

Keywords. E-mail, Inbox, Information, Multimodal, Usability, Visualisation

1. Introduction

Email is the most widely used application nowadays [1, 2] and has become one of the reasons for people buying personal computers [3]. The number of e-mail accounts and messages grows rapidly. For example, it has been estimated that about 31 billion e-mail messages have been sent in 2002 [4]. It has also been shown that the average user gets around 49 e-mail messages a day while high volume users can get more than one hundred [5]. Users usually leave their e-mail messages in the inbox for different purposes such as managing appointments and to-do list. Therefore, e-mail inboxes have become cluttered and difficult to use. As e-mail is used daily in our lives, more usable e-mail clients that help users to browse e-mail easily are needed.

Email data has been graphically visualised by many studies in order to support various issues such as the tasks management. However, graphical presentation of email data has not been significantly considered to improve the usability of email clients. This paper presents an experiment which aimed to investigate whether the usability of email clients can be enhanced by graphically visualising email data. Therefore, two experimental email visualisation tools were developed especially for this experimental programme. They presented email messages based on a dateline alongside other email

information such as the sender's email address and the time. The usability of a typical email client was compared independently to the usability of each experimental email client. The usability were measured objectively by calculating the accomplishment time of experimental tasks and number of errors carried out. This paper concludes with the results of this usability study and highlights some of the future work.

2. Relevant Work

Information visualisation has been employed by many studies to support various issues in e-mail clients. For instance, information visualisation was employed to show message threads, which is the relationship between e-mail messages [6], in the e-mail inbox. Venolia and Neustaedter pointed out that e-mail clients would be more useful if conversation threads were used as the main display for e-mail clients [6]. They presented a mixed-model visualisation that shows the sequence of e-mail messages and reply relationships among the messages of conversation. The users' understanding of message threads was tested and the results showed they were able to understand them. Rohall and colleagues visualised e-mail archives in order to present the relationships between the senders of e-mail messages [1, 5]. Also, the relationships between email messages were presented by displaying the related email messages as a connected tree in different colours. For example, an e-mail message that was coloured purple came from someone outside the recipient's work. Viegas et. al. developed a tool called 'TheMail' for visualising e-mail archives which used the content of e-mail messages to present the change of a relationship over a period of time [7]. This tool presented a series of keywords in columns arranged along a timeline, where each keyword was shown in a different colour. The size of keywords depended on their frequency and distinctiveness. The content of e-mail messages was also used for visualising e-mail archives using self-organising maps that placed similar e-mails close to each other [8].

Email archives were also visualised to support the task management in the email inbox. Yiu et. al. visualised e-mail messages based on the time they were received [9]. E-mail messages were displayed as dots and organized on X and Y axes, where time was presented along the X axis and the sender on the Y axis. 'MailView' is an e-mail tool which visualises e-mail messages depending on the time they are received and presents them as glyphs chronologically enabling users to explore trends over time [4]. Gwizdka developed an email prototype that support the management of pending tasks [10]. Two types of pending tasks arrangements were employed in this prototype automatic and manual arrangement. Sudarsky and Hjelsvold developed a tool that visualised the e-mail inbox depending on a hierarchal nature of domain names in e-mail addresses such as COM and EDU in order help users browse email messages easily [2]. This approach contains two basic hierarchal views, one which is a tree generated from the domain names, and the other is a temporal view which presents the e-mail messages. The tool evaluation results showed a significant improved performance, as well as improved overall preferences [2]. However, the results were brief and the evaluation was informal.

The usability of email clients has not been fully addressed as shown in most of the previous studies. Therefore, this paper shows an experiment that investigates whether the usability of email clients can be improved by visualising email messages.

3. Experimental Platform

An experimental email visualisation platform that provided two email visualisations, which were called LinearVis and MatrixVis, was developed using Microsoft Visual Basic 2005 under a Windows XP platform. It was supported by a database that was designed by Microsoft SQL Server 2005. Both of the visualisation approaches presented email messages based on a dateline together with other email information (e.g., senders' email address, time). Various types of email data were hidden in the two visualisation approaches in order to reduce the graphical complexity in the email inbox and to avoid the visual overload.

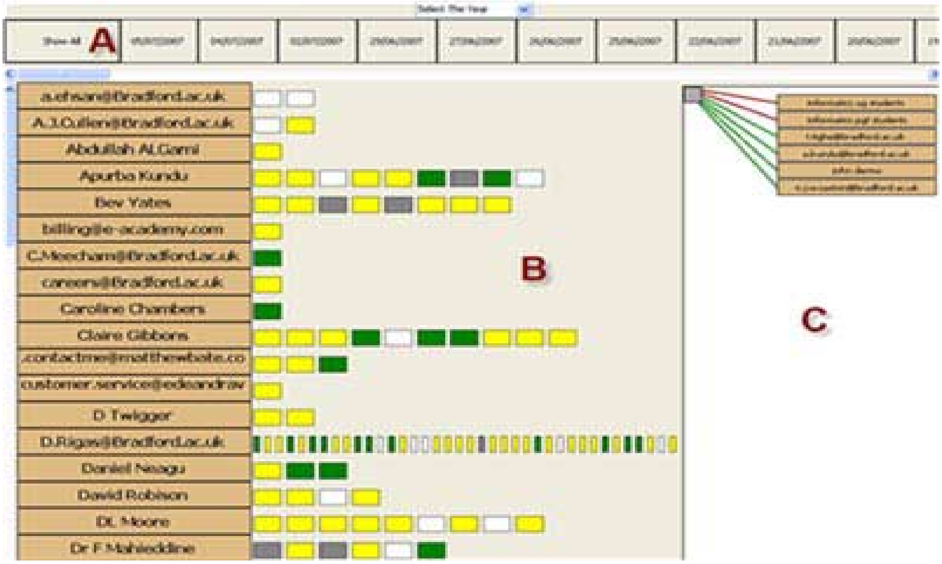


Figure 1The inbox of LinearVis with all email messages presented (A denotes the dateline, B the main view and C the temporal view)

3.1. LinearVis

E-mail messages are visualised in this prototype based on the date and senders' e-mail addresses. Figure1 shows how the inbox in this approach is divided into 3 parts: dateline, main view and temporal view. The top part of the inbox is the dateline, which presents all the dates that contain e-mail messages in chronological order. A drop down menu that contains the previous years (not displayed in the current view) was added in order to reduce the number of presented dates and to reduce the required scrolling operations. All messages in the inbox are presented as squares in the main view, where they are classified based on the alphabetically ordered list of e-mail senders on the left side of the inbox. The size of presented squares depends on the number of e-mail messages sent by the e-mail sender. Moreover, as the number of e-mail messages sent by a sender increases, the size of the presented square will be smaller. Rather than displaying the status (New, Read, Replied, Forwarded) of e-mail messages textually, it is displayed by colours. The unread (New) e-mail messages are displayed in yellow, whereas the read (Opened) messages are displayed in white. The e-mail messages that have been sent as a reply are displayed in green and those that have been forwarded are

displayed in grey. Subject, attachment and priority are hidden in order to reduce the graphical complexity in the main view and to avoid visually overloading users. However, they can be displayed with the content of the e-mail message by clicking on the e-mail message in the main view. The number of displayed messages and addresses in the main view can be reduced by selecting the required date from the dateline.

The recipients of an e-mail message can be shown in the temporal view by moving the mouse cursor over its icon in the main view. Rather than using the traditional textual way of displaying the recipients (TO, CC) of an e-mail message they are presented using colours. All e-mail addresses connected by green lines are those e-mail messages that have been received as a carbon copy (CC), and those received as a normal message (TO) have the e-mail addresses connected by red lines. In order to reduce the amount of presented information on the screen the e-mail addresses that appear in both TO and CC fields are displayed once and connected by blue lines rather than display them twice.

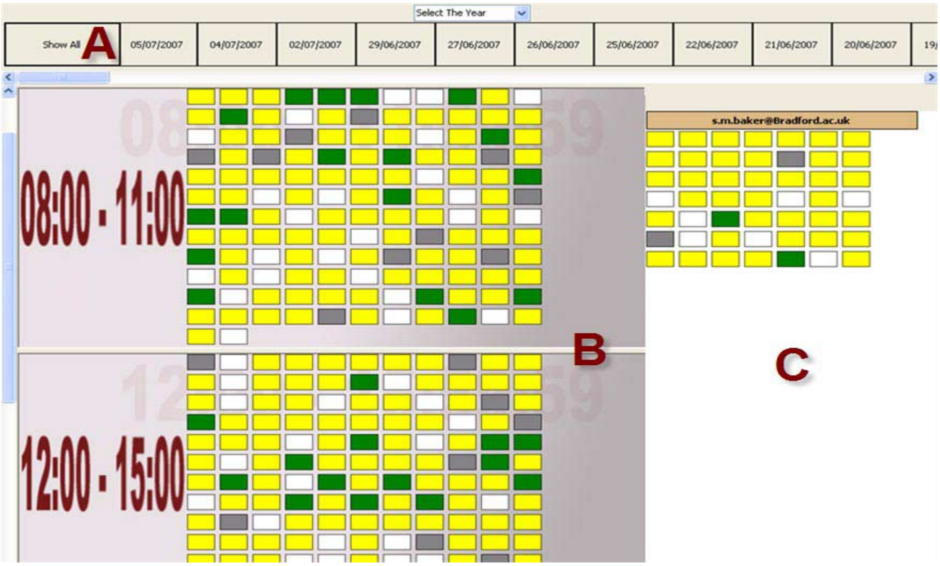


Figure 2 The inbox of MatrixVis with all email messages presented (A denotes the dateline, B the main view and C the temporal view)

3.2. MatrixVis

E-mail messages in this prototype are visualised based on the date alongside the time of receiving messages. Figure 2 shows how the inbox of this visualisation is also divided into 3 main parts: dateline, main view and temporal view. The same dateline of LinearVis is used in this visualisation and is also located at the top part of the inbox. The main view of the MatrixVis inbox is divided into 6 timeslots that present a full day. Each timeslot represents a 4 hours period. E-mail messages are classified into the timeslots according to the time they are received. They are also presented as squares. As the size of the presented squares is fixed, the size of timeslots depends on the number of e-mail messages received in a particular timeslot. Thus, e-mail messages in each of the timeslots are presented chronologically in rows. The maximum number of e-mail messages in each row is 11 e-mail messages (see Figure 2). This means that as

the number of e-mail messages received in a timeslot increases, the size of this timeslot becomes larger. Colours are also used to present the status of the e-mail messages in the same way as LinearVis. The size of the timeslots and the number of e-mail messages displayed in the main view can be reduced by selecting a date from the dateline at the top of the inbox. The content of e-mail messages and the hidden information, such as the subject and the recipients (TO, CC) can be seen by clicking on the required e-mail message in the main view. Senders' e-mail addresses are not presented directly in this visualisation. Therefore, by moving the mouse over an e-mail message, all the e-mail messages sent by the same sender of this message are presented chronologically in the temporal view. The content of e-mail messages presented in the temporal view can be seen by clicking on the required e-mail messages.

Table 1 Experimental Tasks

Task No	Description	Task No	Description
T1	Locating e-mail message by date	T6	Locating e-mail message by sender, status and attachment
T2	Locating e-mail message by status and date	T7	Locating e-mail message by date and attachment name
T3	Locating email message by status, attachment and date	T8	Locating e-mail message by sender and cc
T4	Locating email message by sender and subject	T9	Locating e-mail message by sender, to and cc
T5	Locating e-mail message by sender and priority	T10	Locating e-mail message by status, to and cc

4. Methodology

An experiment was designed in order to test whether information visualisation could significantly enhance the usability of e-mail clients. This experiment is a comparative usability evaluation among one of the well-known e-mail clients, outlook express which is used as the control condition, and the experimental visualisations (i.e. LinearVis, MatrixVis). Thirty postgraduate students from the Department of Computing at the University of Bradford were asked to participate in the experiment. All of them were email users but were not familiar with the provided visualisations. They were asked to perform 10 tasks in each experimental condition in each task they had to locate an e-mail message with the provided relevant information such as the date of receiving, the sender's e-mail address and subject (see Table 1). The experiment was three-conditions, within-subjects design in which each user was asked to perform the experimental tasks in each condition. Users where asked to perform all experimental tasks in each condition before moving to the next one. In order to avoid the learning effects, the order of the conditions was varied between users (counter-balanced) [11]. The e-mail messages were the same under each condition. Users were free to use any of the functions that are offered by Outlook Express such as sorting e-mail messages and searching for an e-mail message whereas they were not able to use these functions in the experimental e-mail visualisations. A five minutes demonstration was given for all users prior performing each condition.

5. Results

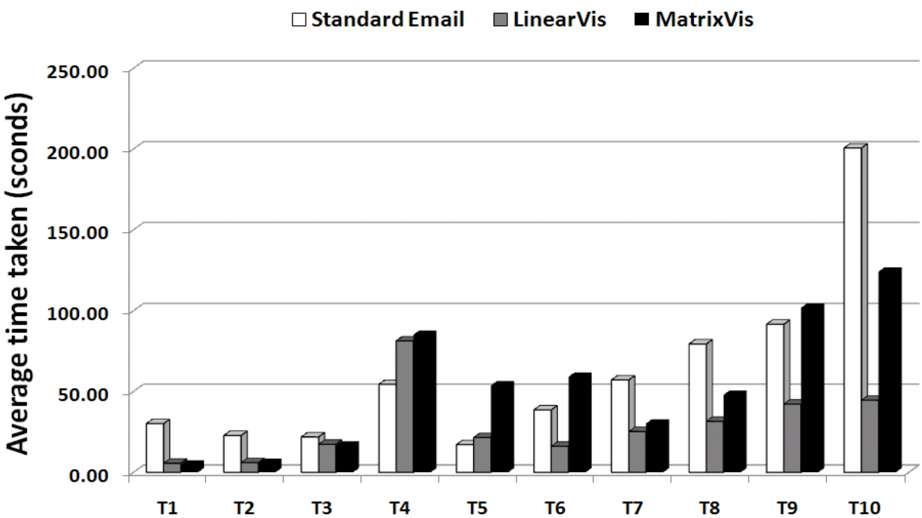


Figure 3 Tasks accomplishment time

5.1. Task Accomplishment Time

The average time taken to complete each experimental task in each condition was calculated in order to test the difference between them. Figure 3 shows the average time taken to complete all experimental tasks in three conditions. The average time taken to complete most experimental tasks in LinearVis was lower than in the control condition. Figure 3 shows that users took longer time to perform task 4 and 5 in LinearVis than in the standard email. This is because subject and priority of email messages were required in these tasks where this type of information is hidden in LinearVis. T-test was performed on the accomplishment time of experimental results in order to test the significance between LinearVis and the standard email. The results indicated that the time taken to complete tasks in LinearVis was significantly reduced ($t18 = 1.7$, $cv= 1.73$, $p<0.05$). Moreover, figure 3 shows that the average time taken to complete most experimental tasks in MatrixVis was reduced except those tasks where users required finding email messages using subject, priority and sender (T4, T5 and T6) as this information is hidden. T-test was also performed to test the difference between MatrixVis and the standard email in terms of accomplishment time. The results showed that MatrixVis has not significantly reduce the time taken to complete the experimental tasks ($t18 = 0.4$, $cv= 1.73$, $p>0.05$).

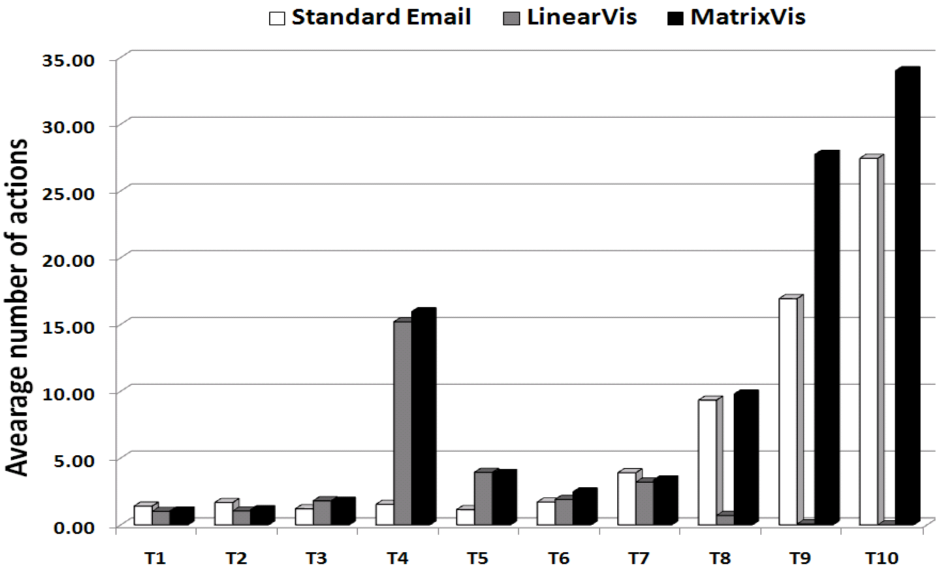


Figure 4 Actions carried out whilst performing tasks

5.2. Number of Actions Carried Out

The number of actions performed by each user to complete the experimental tasks in each experimental condition was added together to produce a total number of actions. Figure 4 shows the average number of actions carried out to perform each experimental task in the three experimental conditions. It shows that users could perform most of the experimental tasks in LinearVis with fewer actions carried out than the standard email. The number of action performed to complete the experimental tasks 4 and 5 in LinearVis was dramatically increased because users needed to open email messages in order to find the required email messages as subject, priority and attachment were hidden. On the other hand, some users could perform the last three tasks (i.e. T8,T9 and T10) without carrying out any action. This is because the recipients of email messages, which are required in these tasks, are presented in the temporal view of LinearVis. T-test was performed on the average number of actions performed by users to test the difference between LinearVis and the standard email. The results indicated that LinearVis was significantly reduced ($t_{18} = 11.4$, $cv = 1.67$, $p<0.05$).

On the other hand, Figure 4 shows that MatrixVis required users to perform more actions than the standard email and LinearVis. In fact, the results showed that MatrixVis is the worst in terms of number of actions. The reason of this can be the extensive use of hidden email data. T-test was also used to test the difference between MatrixVis and the standard email in terms of number of actions. The results showed that the standard email is significantly better than MatrixVis ($t_{18} = -9.2$, $cv = 1.67$, $p<0.05$).

6. Discussion

The analysis of the experimental data showed that LinearVis has significantly reduced tasks accomplishment time and number of actions carried out. On the other hand, MatrixVis has not significantly improved users' performance in terms of time and number of actions. Thus, the main hypothesis that the usability of email clients can be improved by visualising email data was rejected. However, the graphical presentation in LinearVis has confirmed this hypothesis, meaning that the method used for visualising email messages can significantly affect the usability. For instance, the way of presenting email messages in MatrixVis has significantly affected its usability. Dividing the email inbox into multi coordinated view was found very useful for organising email messages such the temporal view and dateline in both experimental emails.

Hiding some email data was not found an effective way for reducing the graphical complexity in the inbox as some of this data can be critical to users. Since users themselves who can decide whether a particular type of email data is important or not, all email data should be presented in the inbox. However, presenting all email data graphically in the inbox will visually overload the users. Therefore, further experiments will to be carried out in order to evaluate the effect of auditory feedback in the LinearVis approach. In these experiments, some email data will be presented aurally in a multi-modal approach in order to reduce the graphical complexity but also avoid hiding email data from the view of the user. Many studies have been performed in the past to test whether multimodal interaction can enhance the usability of user interfaces. It has been showed that it enhances the usability of interfaces in limited size screens such as PDA devices [12-14]. Multimodal feedback has been used for conveying information and the results showed that it could significantly enhance the user's performance [15-17]. Rigas performed a series of experiments that investigated the use of auditory stimuli to communicate data in information systems [18-20]. The results showed that auditory stimuli helped to communicate information that users received via the visual channel.

7. Conclusion

The experiment described in this paper showed that the graphical presentation of email data, such as the one in LinearVis, can be successfully used to improve the usability of e-mail clients. It was also found that the method used for visualising email messages is crucial factor on the usability. The experimental results indicated that hiding email data in order to reduce the graphical complexity could significantly reduce the usability of e-mail as the hidden data could be critical to some users. Previous studies showed that auditory stimuli complemented and aided users to successfully interpret incomplete visual information. Therefore, further experiments will be carried out in order to investigate the employment of auditory feedback in the graphical representation of email data.

References

[1] Rohall, S.L., *Redesigning Email for the 21 stCentury Workshop Position Paper.* CSCW 2002 Workshop: Redesigning Email for the 21st Century. New Orleans, LA, 2002.

[2] Sudarsky, S. and R. Hjelsvold, *Visualizing electronic mail.* Information Visualisation, 2002. Proceedings. Sixth International Conference on, 2002: p. 3-9.

[3] Whittaker, S., V. Bellotti, and P. Moody, *Revisiting and Reinventing Email.* Special Issue of Human-Computer Interaction, 2005. **20**(1).

[4] Frau, S., J.C. Roberts, and N. Boukhelifa, *Dynamic Coordinated Email Visualization.* WSCG05: p. 187–193.

[5] Rohall, S., et al., *Email Visualizations to Aid Communications.* IEEE Symposium on Information Visualization (InfoVis), 2001.

[6] Venolia, G.D. and C. Neustaedter, *Understanding sequence and reply relationships within email conversations: a mixed-model visualization.* Proceedings of the conference on Human factors in computing systems, 2003: p. 361-368.

[7] Viegas, F.B., S. Golder, and J. Donath. *Visualizing email content: Portraying relationships from conversational histories*. 2006. Montreal, QC, Canada: Association for Computing Machinery, New York, NY 10036-5701, United States.

[8] Nürnberger, A. and M. Detyniecki, *Externally Growing Self-Organizing Maps and Its Application to E-mail Database Visualization and Exploration.* Applied Soft Computing, 2006. **6**(4): p. 357-371.

[9] Yiu, K., et al., *A time-based interface for electronic mail and task management.* Proc. HCI International'97, 1997. **2**: p. 19-22.

[10] Gwizdka, J., *Reinventing the inbox: supporting the management of pending tasks in email.* Conference on Human Factors in Computing Systems, 2002: p. 550-551.

[11] Dix, A.J., *Human-computer interaction*. 3rd ed. 2004, New York; London: Prentice-Hall. xvi,570p.

[12] Grasso, M.A., D.S. Ebert, and T.W. Finin, *The Integrality of Speech in Multimodal Interfaces.* ACM Transactions on Computer-Human Interaction, 1998. **5**(4): p. 303-325.

[13] Vo, M.T. and A. Waibel, *A Multimodal Human-Computer Interface: Combination of Speech and Gesture Recognition.* Adjunct Proceedings of INTERCHI. **93**.

[14] Dusan, S., G.J. Gadbois, and J. Flanagan, *Multimodal Interaction on PDA's integrating Speech and Pen Inputs.* Proc of. EUROSPEECH'03, 2003.

[15] Vitense, H.S., J.A. Jacko, and V.K. Emery, *Foundation for improved interaction by individuals with visual impairments through multimodal feedback.* Universal Access in the Information Society, 2002. **2**(1): p. 76-87.

[16] Rigas, D. and A. Ciuffireda, *Multi-Modal Web-Browsing An empirical Approach To Improve the Browsing Process of Internet Retrieved Results* SIGMAP 2006 - INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING AND MULTIMEDIA APPLICATIONS, 2006: p. 296-276.

[17] Rigas, D. and M. Alsuraihi, *A Toolkit for Multimodal Interface Design: An Empirical Investigation.* LECTURE NOTES IN COMPUTER SCIENCE, 2007. **4552**: p. 196.

[18] Rigas, D. and J. Alty, *The rising pitch metaphor: an empirical study.* International Journal of Human-Computer Studies, 2005. **62**(1): p. 1-20.

[19] Rigas, D.I., J.L. Alty, and F.W. Long, *Can music support interfaces to complex databases.* EUROMICRO-97, New Frontiers of Information Technology, 1997: p. 78–84.

[20] Rigas, D.I. and D. Memery, *Utilising audio-visual stimuli in interactive information systems: a two domain investigation on auditory metaphors.* Information Technology: Coding and Computing, 2002. Proceedings. International Conference on, 2002: p. 190-195.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-250

Designing a multi-modal affective knowledge-based user interface: combining empirical studies

Maria VIRVOU[1], George TSIHRINTZIS[1] ,Efthymios ALEPIS[1], Ioanna-Ourania STATHOPOULOU[1]

[1]Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou St., 18534 Piraeus, Greece

E-mail: mvirvou@unipi.gr; geoatsi@unipi.gr; talepis@unipi.gr; iostath@unipi.gr

Abstract: In this paper, we present and discuss three empirical studies that we have conducted involving human subjects and human observers concerning the recognition of emotions from audio-lingual visual-facial and keyboard-evidence modalities. Many researchers agree that these modalities are complementary to each other and that their combination can improve the accuracy in affective user models. However, there is a shortage of research in empirical work concerning the strengths and weaknesses of each modality so that more accurate recognizers can be built. In our research, we have investigated the recognition of emotions with respect to 6 basic emotional states, namely *happiness*, *sadness*, *surprise*, *anger* and *disgust* as well as the emotionless state which we refer to as *neutral*. We have concluded that, in cases where a single modality may be deficient in providing emotion recognition evidence, the recognition process can be supported and complemented by the other modalities.

1. Introduction

Recently, the recognition of emotions of users while they interact with software applications has been acknowledged as an important research topic. How people feel may play an important role on their cognitive processes as well [11]. Thus, the whole issue of human-computer interaction has to take into account the feelings of the users. Picard [14] points out that one of the major challenges in affective computers is to try to improve the accuracy of recognizing people's emotions. Improving the accuracy on emotion recognition may imply the combination of many modalities in user interfaces. Indeed, human emotions are usually expressed in many ways. For example, as we articulate speech we usually move the head and exhibit various facial emotions [12]. Ideally, evidence from many modes of interaction should be combined by a computer system so that it can generate as valid hypotheses as possible about users' emotions. This view has been supported by many researchers in the field of human computer interaction [13], [3], [4]. However, progress in emotion recognition based on multiple modalities has been quite slow. Although several approaches have been proposed to recognize human emotions based on facial expressions or speech, relatively limited work has been done to fuse these two and other modalities to improve the accuracy and robustness of the emotion recognition system [2].

In view of the above, it is our aim to combine modalities in order to improve the accuracy of emotion recognition. In [1], we describe briefly a system that we have already constructed for combining two modalities, namely keyboard stroke pattern and audio-lingual information. In the present paper, we focus on the incorporation of a third modality, namely visual-facial information. Towards combining the three modalities, we had to determine first the extent to which these two different modalities can provide emotion recognition from the perspective of a human observer. Moreover, we had to specify the strengths and weaknesses of each modality. For these purposes, we conducted corresponding empirical studies.

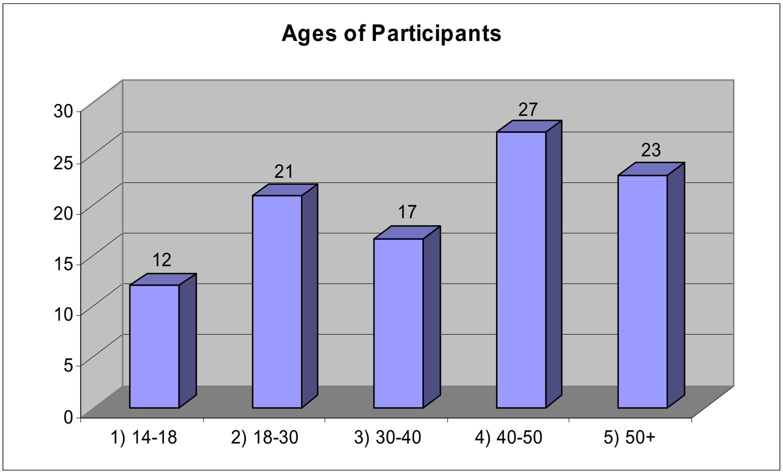
These empirical studies provide the basis towards the combination of modalities into the affective user modeling component of our tri-modal system. They are also expected to be useful to other researchers, as there is a shortage of such empirical studies in the literature. Indeed, the most relevant research work is the empirical study of De Silva *et al* on humans' ability to recognize emotions [4]. However, De Silva *et al* focus on the audio analysis of voice signals in terms of pitch and volume of voice rather than lingual keywords that convey affective information. On the other hand, in our research we include the lingual aspect of users' spoken words on top of the pitch and volume of voice. Moreover, we compare the audio-lingual results with corresponding results from the other two modalities to determine the modality which conveys more information for human observers. Our study concentrates on six basic emotions, namely *happiness*, *sadness*, *surprise*, *anger* and *disgust*, as well as the emotionless state which we refer to as *neutral*.

In this paper, we present all of the empirical studies and show and discuss results from their comparison. This paves the way to combination methods that take into account the human observers' performances in the recognition of emotions.

2. Empirical studies for audio-lingual and keyboard stroke pattern emotion recognition

The empirical study involved 100 users (male and female) of various educational background, age and levels of computer experience. People's behavior while doing something may be affected by several factors concerning their personality, age, experience, etc. For example, experienced computer users may use a keyboard more often than novice users as a mode of interaction. On the other hand, younger people may prefer different approaches in interacting with computers, such as audio-lingual interaction. Thus, for the purpose of analyzing the results of our empirical study while taking into account important characteristics of users, we categorized the users into several groups, as presented below.

Figure 1.a illustrates the distribution of participants in the empirical study in terms of age. In particular, 12% of participants were under the age of 18, while approximately 21% of them were between the ages of 18 and 30. The remaining 66% of the participants were over the age of 30. Similarly, figure 1.b illustrates the distribution of participants in the empirical study in terms of their computer knowledge level.



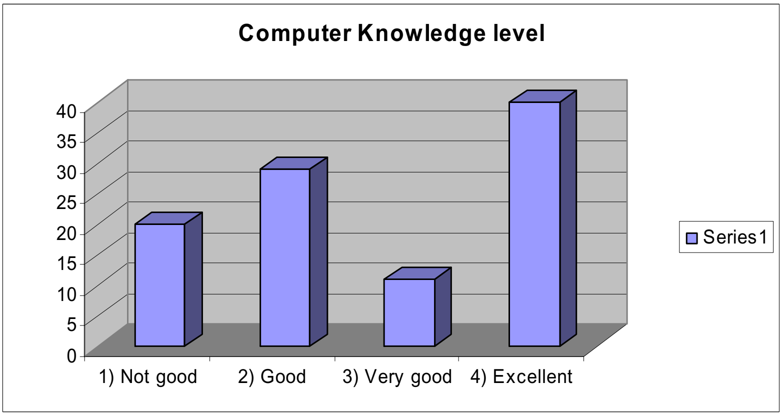


Figure 1.a Ages of participants
Figure 1.b Computer knowledge level of the participants

2.1 Keyboard stroke pattern analysis

The basic aim of this experiment was to capture data with the keyboard. A monitoring component recorded the actions of users while they interacted with an educational application. Data were stored into a database together with corresponding video clips of the users. After done with using the educational application, participants were asked to watch the corresponding video clip of their personal interaction and to determine in which situations they where experiencing changes in their emotional state. Then, they associated each change in their emotional state with one of the six basic emotion states of our study and the data was recorded with a time stamp.

As a next step, the collected transcripts were then given to 20 human experts-observers who were asked to perform emotion recognition with regard to the six emotion states, namely happiness, sadness, surprise, anger, disgust and neutral. All the

human experts held a first and/or higher degree in Computer Science. In the case of the keyboard empirical study, human experts-observers analyzed the data corresponding to keyboard stroke patterns only. Thus, they were asked to watch the video tape and were also given a print out of what the user had written as well at the exact time of each event as it was captured by the user monitoring component. Correct or wrong answers, as well as the consequences of these events, were analyzed as long as these events involved only the keyboard modality. Frequent use of backspace and other basic keyboard buttons was also recorded and associated with specific emotional states.

While using the keyboard modality, the empirical study revealed that when participants were nervous the possibility of making mistakes increased rapidly. This was also the case when they had negative feelings. Mistakes in typing were followed by many backspace-key strokes and concurrent changes in the emotional state of the user at a percentage of 82 %. Yet users under 20 years old and users who over 20 years old with low educational background seemed to be more prone to making even more mistakes as a consequence of an initial mistake and lose concentration while interacting with the educational application (67%). From the analysis of the data, the conclusion is drawn that when the participants were angry the rate of mistakes increased, the rate of their typing become slower (62 %). On the contrary, when they were happy they typed faster (70 %) and the keystrokes become harder (65 %). The pressure of the user's fingers on the keyboard would give a further indication in recognizing emotional states of users, but could not be measured in our experiments. Similar observations in the interaction via the keyboard were reported with regards to the emotions of boredom and anger.

Table 1 illustrates the percentages of successful emotion recognition by human experts concerning the participants' emotional states and the keyboard mode of interaction. These percentages result from the comparison of the recognized emotional states by the human experts and the actual emotional states as recorded by the participants themselves.

Table 1. Percentages of successful emotion recognition by human experts.

Emotional state	Percentage of recognition by human experts (keyboard modality)
Neutral	65%
Happiness	60%
Sadness	57%
Surprise	5%
Anger	74%
Disgust	4%

2.2 Audio-lingual analysis

The participants were asked to interact with an educational application which incorporated a user monitoring component. Audio data, provided orally by the user, were recorded into a database together with corresponding video clips of the users. After done with using the educational application, participants were asked to watch the corresponding video clip of their personal interaction and to determine in which situations they where experiencing changes in their emotional state. Then, they associated each change in their emotional state with one of the six basic emotion states of our study and the data was recorded with a time stamp.

Next, the collected transcripts were given to 10 human experts-observers who were asked to perform emotion recognition with regard to the six emotion states, namely happiness, sadness, surprise, anger, disgust and neutral. In this study, human experts-observers analyzed the data from the audio-lingual input only.

The audio-lingual empirical study gave important results about the strengths and weaknesses of emotion recognition that is based on the audio-lingual modality. The human experts' recognition rates of the six emotions showed that some emotions are easily recognised by the audio-lingual modality. Such an emotion is sadness. On the other hand, other emotions are more difficult to recognise by the audio-lingual modality. Such an emotion is surprise. The results from the empirical study could be either confirmed be the recognition of the users' emotions in real-time (by using the user modelling component), or in some cases take specific results into reconsideration. Combining the two phases of the empirical study, we came up with a database constructed from the users' oral input and containing words, phrases and exclamations. At the same time, changes in voice volume and voice pitch were recorded in relation with the oral input.

Table 2 illustrates the results of the empirical study in terms of the audio-lingual input via microphone and the six basic recognized emotions (neutral, happiness, sadness, surprise, anger, disgust) by human expert-observers. For each emotion, we observe the percentages of the users' oral reaction or the absence of audio input. Furthermore, we observe the changes in the users' voice (volume and pitch) while saying a word or phrase or saying an exclamation in each emotional situation. For example, a user in surprise may have said an exclamation (58 %) rather than having spoken a word (24 %). This action is recognised to a degree of 66 % as accompanied by an increase in the user's volume. In addition, the audio-lingual empirical study supplied us with a significant number of words, phrases and exclamations, which are used in the creation of an "active grammar", with specific words and phrases for emotion recognition.

Table 2. Basic Emotional States through Microphone - Empirical Study Results.

	Say an exclamation		Say a certain word or phrase		Say nothing	
Emotions	Change in volume	Change in pitch	Change in volume	Change in pitch	Change in volume	Change in pitch
Neutral	6%		22%		72%	
	45%	18%	37%	12%		
Happiness	31%		45%		24%	

	40%	37%	55%	25%		
Sadness	8%		28%		64%	
	52%	34%	44%	14%		
Surprise	58%		24%		18%	
	66%	23%	60%	21%		
Anger	39%		41%		20%	
	62%	58%	70%	62%		
Disgust	50%		39%		11%	
	64%	43%	58%	38%		

We may observe that the neutral emotion and sadness could not be recognized easily in this modality because users did not express themselves orally when they were in these emotional states. Users expressing the emotions of surprise and disgust usually said an exclamation (58 % and 50 %, respectively) while happy users and users in anger would likely say a word or phrase contained in our emotional database of words and phrases (45 % and 41 %, respectively). Particularly, with regards to the emotions of surprise and anger, users would increase the volume of their voice while saying something, while in anger and disgust we could observe the significant changes in the users' voice pitch.

3. Empirical Studies for Visual-Facial Emotion Recognition

To obtain a preliminary idea of facial expression classification by humans, we developed a preliminary questionnaire in which we asked 300 study participants to classify the facial expressions that appeared in 36 images. Each participant could choose from 11 of the most common facial expressions, such as: "angry", "happy", "neutral", "surprised", or specify any other expression that he/she thought appropriate. Next, the participant had to decide which emotion he/she thought that the facial expression indicated.

Our dataset consisted of 3 subsets of images, as typically depicted in Table 3, that is:

- various images of individuals placed in a background and forming a facial expression,
- a sequence of facial expressions of the same person without a background, and
- face images of different persons without a background.

From the results of the questionnaire, we observed that the "surprised" expression was the one most easily recognized, as the corresponding error rate of 22% was the lowest among all error rates. The "happy" and "neutral" expressions were recognized with corresponding error rates of 30% and 35%, respectively. The expression recognized with the highest difficulty was the "sad" expression, as its corresponding error rate reached 88%. The "angry" and "disappointed" expressions had a corresponding error rate of 80% and 76%, respectively. These are summarized in Figure 2.

Table 3. Typical face image subsets in our questionnaire

	The three sets of our questionnaire
1st Set	
2nd Set	
3rd Set	

Clearly, the facial expression classification task in images is quite challenging. The reasons why humans could not achieve low classification error rates for specific expressions such as “angry”, “sad” and “disappointed”, may be found in the fact that these expressions seem to differ significantly from person to person or some people may be too shy to form them clearly.

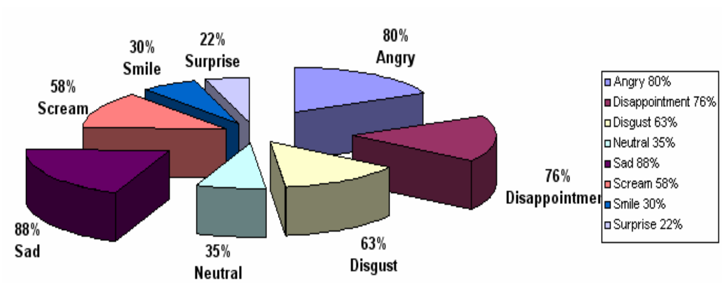


Figure 2. Error rates in recognizing the expressions in our preliminary questionnaire

4. Combination of the results from the three modes of interaction

The three modes of affective human-computer interaction (audio-lingual, keyboard stroke pattern and visual-facial) are complementary to a high degree. Moreover, even in cases where all three modalities show comparable percentages of successful emotion recognition, a combination of the three leads to gains in the probability of accurate emotion recognition.

The purpose of the paper is to present concisely empirical studies concerning emotion recognition with audio-lingual, keyboard stroke pattern and visual-facial modalities and draw conclusions regarding the combination of these modalities. This

will allow us to set the empirical basis for creating an improved *tri-modal affective human-computer interaction system*. Eventually, the empirical data from the three modalities will be combined with a multi-criteria decision theory in which each mode will be viewed as a criterion.

The percentages of emotion recognition for each emotion and for each modality will be analyzed and then used as weights in order to improve the accuracy of the tri-modal system and determine the prevailing emotion. The combined emotion recognition process is illustrated compactly in Figure 3.

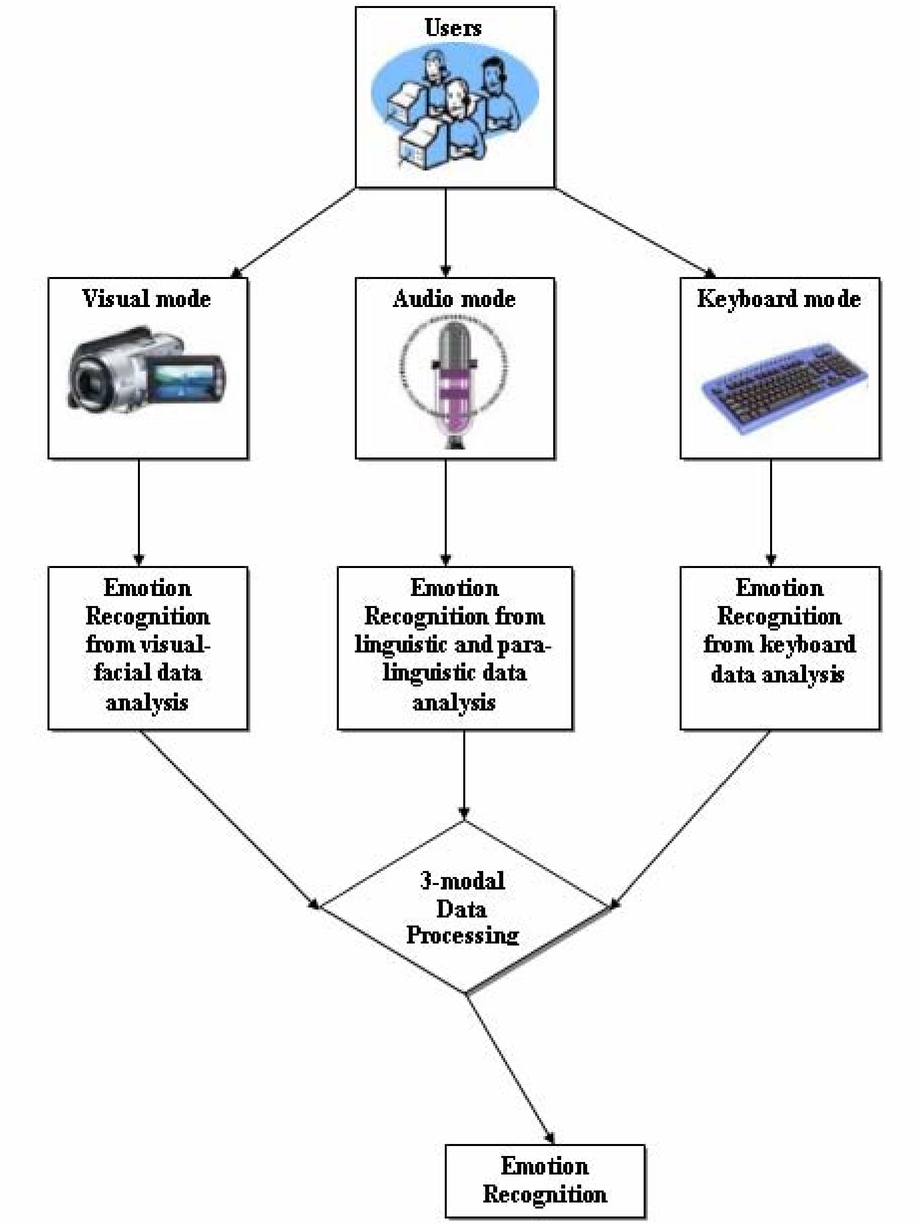


Figure 3. Emotion recognition using three modes of interaction

5. Conclusions

In this paper we have described and discussed three empirical studies that concern the audio-lingual the visual-facial recognition and the recognition through the keyboard of human users' emotions from the perspective of human observers. These modalities are complementary to each other and, thus, can be used in a multi-modal affective computer system that can perform affect recognition taking into account the strengths and weaknesses of each modality.

From the empirical studies we found that certain emotion states such neutral and surprise are more clearly recognized from the visual-facial modality rather than the audio-lingual. Other emotion states, such as anger and disgust are more clearly recognized from the audio-lingual and the keyboard-stroke modalities respectively, rather than the visual-facial.

There is ongoing research of construction of an affective user interface that will use the different modalities as criteria for recognition of emotions and will use the results of performances for each modality as the basis for the specification of weights. This and other related work is going to be presented in a future publication.

6. Acknowledgements

This research was supported by the General Secretariat of Research and Technology, under the auspices of the PENED-2003 basic research program. Travel funds to present this work were provided by the EPEAEK program.

7. References

1. Alepis, E., Virvou, M., Kabassi, K.: Affective student modeling based on microphone and keyboard user actions, Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies (ICALT), Kerkrade, The Netherlands (2006) 139 - 141
2. Busso, C., Deng , Z., Yildirim, S., Bulut, M., Lee, C.M., Kazemzadeh, A., Lee, S., Neumann, U., Narayanan, S.: Analysis of emotion recognition using facial expressions, speech and multimodal information, Proceedings of the 6th international conference on Multimodal interfaces, State College, PA, USA (2004) 205 - 211
3. Chen, L.S., Huang, T.S., Miyasato, T., Nakatsu, R.: Multimodal human emotion/expression recognition, in Proc. of Int. Conf. on Automatic Face and Gesture Recognition, IEEE Computer Soc., Nara, Japan (1998)
4. De Silva, L., Miyasato, T., Nakatsu, R.: Facial emotion recognition using multimodal information, in Proc. IEEE Int. Conf. on Information, Communications and Signal Processing (ICICS'97) (1997) 397-401
5. Ekman, P.: Emotion in the human face, New York: Cambridfe University Press (1982)
6. Ekman, P.: Basic Emotions, In T. Dalgleish and T. Power (Eds.) The Handbook of Cognition and Emotion Pp. 45-60. Sussex, U.K.: John Wiley & Sons, Ltd (1999)
7. Ekman, P.: An argument for basic emotions. Cognition and Emotion 6, (1992) 169-200

8. Ekman, P., Davidson, R.J.: The Nature of Emotion. Fundamental Questions, Oxford University Press Inc. (1994)
9. Ekman, P., Friesen, W.V.: Facial action coding system: A technique for the measurement of facial movement. Palo Alto, Calif.: Consulting Psychologists Press, (1978)
10. Ekman, P., Friesen, W.V.: Unmasking the face. A guide to recognizing emotions from facial clues. Englewood Cliffs, New Jersey: Prentice-Hall, (1975)
11. Goleman, D.: Emotional Intelligence, Bantam Books, New York (1995)
12. Graf, H.P., Cosatto, E., Strom, V., Huang, F.J.: Visual Prosody: Facial Movements Accompanying Speech, Proceedings of the fifth IEEE International Conference on Automatic Face and Gesture Recognition (2002) 381-386
13. Pantic, M., Rothkrantz, L.J.M.: Toward an affect-sensitive multimodal human-computer interaction. Vol. 91, Proceedings of the IEEE (2003) 1370-1390
14. Picard, R.W.: Affective Computing: Challenges, Int. Journal of Human-Computer Studies, Vol. 59, Issues 1-2, (2003) 55-64
15. Russell, J.A.: Is there universal recognition of emotion from facial expression? Psychological Bulletin, 95, 102-141 (1994)
16. Russell, J.A., Fernandez-Dols, J.M.: The Psychology of Facial Expression. Cambridge University Press: New York, (1997)

This page intentionally left blank

Knowledge Technologies for Semantic Web

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-263

Integrating a Domain Ontology Development Environment and an Ontology Search Engine

Takeshi MORITA [a,b,1] Noriaki IZUMI [c] Takahira YAMAGUCHI [a]
[a] *Graduate School of Science and Technology, Keio University*
[b] *Research Fellow of the Japan Society for the Promotion of Science*
[c] *National Institute of Advanced Industrial Science and Technology*

Abstract. In order to reduce the cost of building domain ontologies manually, in this paper, we propose a method and a tool named DODDLE-OWL for domain ontology construction reusing texts and existing ontologies extracted by an ontology search engine: Swoogle. In the experimental evaluation, we applied the method to a particular field of law and evaluated the acquired ontologies.

Keywords. ontology development environment, ontology search engine

1. Introduction

In order for Semantic Web to be achieved, ontologies are constructed in distributed places or domains, and then mapped together with each other. However, the large cost for domain ontology development and maintenance is a bottleneck for this goal. In order to resolve these issues, many studies are focusing on using knowledge engineering techniques, natural language processing, and data-mining methods [1,2,3] to make possible automatic and semi-automatic domain ontology construction from existing information resources, such as texts and general ontologies. Although these approaches reduce the cost of constructing domain ontologies, there are still several issues left. Furthermore, these approaches for constructing domain ontologies can not support the reuse of existing ontologies which describe specific domains.

Currently, the number of domain ontologies on the web has increased. In order to find ontologies on the web, ontology search engines have been and are still being developed. Swoogle [4] is an ontology search engine which uses `OntoRank` to rank the ontologies and `TermRank` to rank classes and properties by their popularity. From the viewpoint of domain ontology construction, not all popular ontologies are reusable. In order to appropriately find reusable ontologies, a mechanism to find existing ontologies for the target domain is necessary.

[1]Corresponding Author: Takeshi MORITA, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama-shi, 223-8522 Japan; E-mail: t_morita@ae.keio.ac.jp.
This work was supported by Grant-in-Aid for JSPS Fellows (19-9818).

In order to resolve the above issues, in this paper, we propose a method and a tool for domain ontology construction reusing texts and existing ontologies extracted by an ontology search engine: Swoogle. In our former study [5], we developed a tool named DODDLE-OWL (a Domain Ontology rapiD DeveLopment Environment - Web Ontology Language extension) to construct domain ontologies in OWL. DODDLE-OWL supports the semi-automatic construction of both taxonomic and other relationships in domain ontologies. However, the former DODDLE-OWL could not reuse existing ontologies on the web. We considered that if the ontologies for a target domain existing on the web can be reused, the costs for refining semi-automatically generated ontologies will be reduced. In order to reuse the existing ontologies, we extend the following functions into DODDLE-OWL: Acquiring the existing ontologies from the web for a target domain with Swoogle, gathering ontological elements from the existing ontologies, ranking the ontologies retrieved, and reusing the ontologies to construct taxonomic relationships using ontology alignment and other relationships.

This paper is structured as follows. In Section 2 and Section 3, we first present related works and the design of DODDLE-OWL. We then show the implementation of our proposed method in Section 4. In Section 5, we present experimental evaluation. Finally, we conclude this paper and point out the future work in Section 6.

2. Related Works

2.1. Ontology Development Environment

In order to reduce the high cost for building domain ontologies manually, automatic and semi-automatic methods have been proposed using knowledge engineering techniques, natural language processing, and data-mining methods. Most of the tools (e.g. Text2Onto [1], OntoLT [2], DODDLE-II [3]) support the construction of both taxonomic and other relationships reusing existing information resources, such as texts and general ontologies (e.g. WordNet).

Concept drift management is the main issue when reusing general ontologies for the construction of taxonomic relationships in a domain ontology. Concept drift means that some parts of hierarchical relations are counterchanged between a general ontology and a domain ontology. Ontologies generated automatically which contain concept drifts may cause confusion for domain experts. DODDLE-II, which is our former study, provides facilities to manage the concept drift. However, the domain experts still face the high costs for refining the automatically generated ontology.

2.2. Ontology Search Engine

In order to reuse existing domain ontologies, the user needs to find domain ontologies for the target domain. OntoSelect [6] supports the search, selection, and browsing of ontologies on the web. Our proposed method for ranking existing ontologies is similar to the method for selecting appropriate ontology in [6].

Swoogle [4] is an ontology search engine which indexes over 10,000 ontologies as of 2007. Swoogle can find classes and properties in ontologies, and also find implicit links and relations which are not defined in the ontologies. Swoogle also provides REST

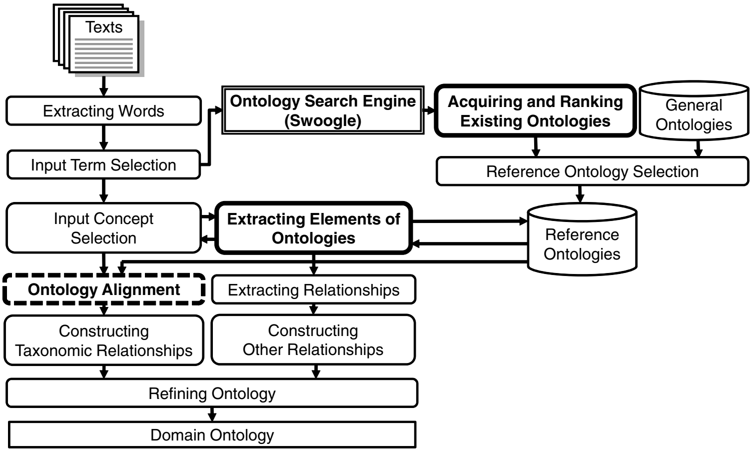


Figure 1. The system flow of DODDLE-OWL using Swoogle

(Representational State Transfer) web-service interface for machine agents to avoid html-scraping. Our system uses this web-service interface to acquire existing ontologies for a target domain. Swoogle uses `OntoRank` to rank ontologies and `TermRank` to rank classes and properties by their popularity as in the `PageRank` algorithm. From the view-point of domain ontology construction, not all popular ontologies are reusable. In order to find reusable ontologies appropriately, the mechanism to find existing ontologies for the target domain is necessary.

Watson [7] is a gateway to the Semantic Web: it collects, analyzes and gives access to ontologies and semantic data available online. Watson also provides Watson API and Web services for machine agents. In the future, we would like to try to use the Watson API to acquire existing ontologies.

3. A Domain Ontology Construction Method Supported by an Ontology Search Engine

3.1. Designing a Domain Ontology Development Environment

In this Section, we describe a method and a tool named DODDLE-OWL (a Domain Ontology rapiD DeveLopment Environment - Web Ontology Language extension) for domain ontology construction reusing texts and existing ontologies extracted by an ontology search engine. DODDLE-OWL supports the semi-automatic construction of both taxonomic and other relationships in domain ontologies. Figure 1 shows the system flow of DODDLE-OWL. The input of DODDLE-OWL is texts and the output of DODDLE-OWL is a domain ontology which consists of taxonomic relationships and other relationships. First, DODDLE-OWL extracts noun and verb phrases from the texts using morphological analyzer. The user selects important terms (input terms) for the domain from the extracted words. DODDLE-OWL acquires existing ontologies using Swoogle based on the input terms and ranks them. Referring to the ranking, the user selects existing ontologies which are reusable to construct a domain ontology. The selected ontologies are

called reference ontologies, and become the basis of the domain ontology. Depending on the target domain, existing ontologies may not be acquired. To cover this situation, DODDLE-OWL uses general ontologies (WordNet and EDR) as reference ontologies. Then, DODDLE-OWL extracts concepts which have the input term as their label from the reference ontologies. If an input term has ambiguity, the user selects the most appropriate concept for the domain from the extracted concepts. We call the selected concepts as input concepts. DODDLE-OWL then automatically extracts paths including input concepts from the reference ontologies, merges them, and constructs the initial class and property hierarchy. It is difficult to directly merge the extracted paths from the heterogeneous existing ontologies on the web due to the difference between structures of the upper concept hierarchies. Therefore, identification of similar concepts using ontology alignment is necessary. In order to support the construction of other relationships for the domain ontology, DODDLE-OWL extracts from the reference ontologies, the properties which define the input concepts as their `rdfs:domain` and `rdfs:range`. If the properties are not defined in the reference ontologies, DODDLE-OWL extracts concept pairs by co-occurrency based statistic methods. The details of the methods are described in [3]. Finally, the user refines the initial ontology and the domain ontology is constructed.

In order to support the construction of the domain ontology using Swoogle, we need to extend the following functions to our former study [5]: Acquiring existing ontologies for a target domain using Swoogle, extracting elements from the existing ontologies, ranking the existing ontologies, and reusing the ontologies to construct taxonomic relationships using ontology alignment and other relationships. The rest of this Section will describe these functions.

3.2. Acquiring and Ranking Existing Ontologies using an Ontology Search Engine

DODDLE-OWL acquires and ranks existing ontologies for a target domain using Swoogle by the following procedures depicted in Figure 2. The detail of the implementation is described in Section 4.1.

1. Acquiring the classes and properties which have the input terms (important terms for a domain from domain specific texts) as their URI's local name or as the value of `rdfs:label` property. These classes and properties are named input concepts.
2. Acquiring the properties which have the classes acquired from step 1 as their value of `rdfs:domain` or `rdfs:range` property.
3. Acquiring the value of `rdfs:domain` and `rdfs:range` of the properties which are acquired from step 1 and 2.
4. Acquiring the ontologies which define the classes and properties acquired from step 1 through 3.
5. Referring the ontologies acquired from step 4, removing the properties which are acquired from step 1 and 2, where the value of `rdfs:domain` or `rdfs:range` is neither the input concept nor the upper concept of the input concept.
6. Gathering ontological elements from the acquired ontologies using templates described in SPARQL.
7. Ranking the acquired ontologies mainly using the ratio of input concept in the ontology.

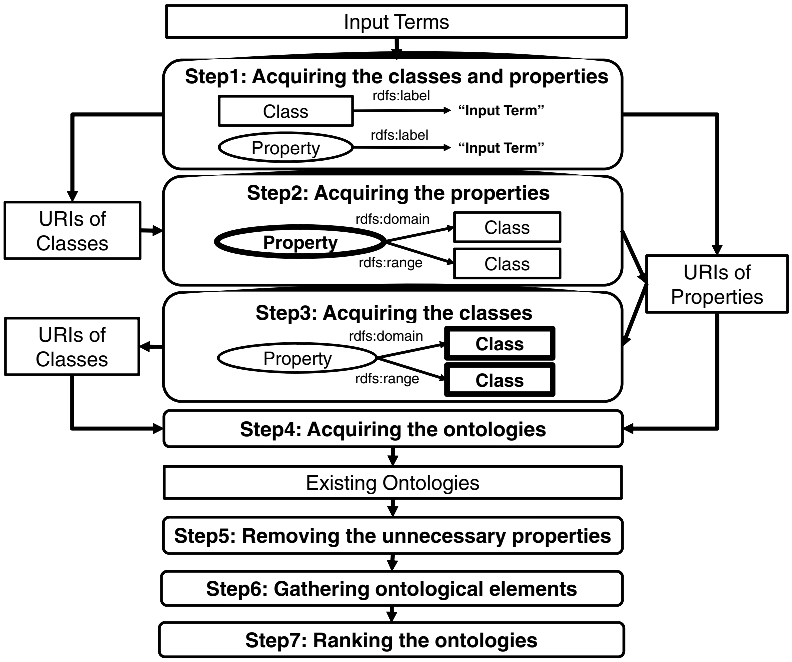


Figure 2. The procedure flow for acquiring and ranking existing ontologies for a target domain using Swoogle

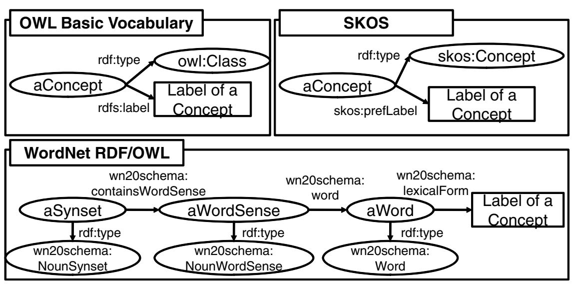


Figure 3. Difference of the labels of concepts among OWL Basic Vocabulary, SKOS, and the WordNet RDF/OWL scheme

3.3. Extracting Elements from Existing Ontologies

In order to reuse existing ontologies for domain ontology construction, it is necessary to extract reusable elements from existing ontologies. DODDLE-OWL supports the construction of taxonomic and other relationships in the domain ontology. The elements constructing taxonomic and other relationships are concepts (classes and properties), labels of concepts, descriptions of concepts, super-subrelations, and other relations. Concept is absolutely essential for the domain ontology construction. Labels of concepts are necessary to find concepts related to input terms. Descriptions of concepts are necessary for word sense disambiguation. Super-subrelations are necessary to construct tax-

onomic relationships. Other relations include the definitions of the properties and their `rdfs:domain` and `rdfs:range`. Other relations are necessary to construct other relationships. Ontology languages (e.g. RDFS, DAML, and OWL) provide the vocabularies to define the above elements of ontology.

In Swoogle, the user can search ontologies based on the vocabularies provided by RDFS, DAML, and OWL. Most of the existing ontologies are described in basic vocabularies of RDFS, DAML, and OWL. However, some general ontologies and thesauruses are described in other schemes. WordNet RDF/OWL [2] provides a standard conversion of WordNet for direct use by Semantic Web application developers. The WordNet RDF/OWL schema is different from the OWL basic vocabularies. SKOS (Simple Knowledge Organisation System) [3] provides a model for expressing the basic structure and the content of concept schemes. Some thesauruses are converted to SKOS. SKOS is also different from the OWL basic vocabularies. Figure 3 shows the difference of the labels of concepts among OWL basic vocabulary, SKOS, and the WordNet RDF/OWL scheme. In Figure 3, we regard a synset in WordNet as a concept.

From the viewpoint of domain ontology construction reusing existing ontologies in various schemes, including thesauruses, we use five types of templates for extracting the elements of ontology described in SPARQL query language for RDF [4]. The five types of templates each extract one of the following elements: Classes, properties, labels and descriptions, super-subrelations, and other relations. The detail of the implementation is described in Section 4.2.

3.4. Ranking Existing Ontologies

DODDLE-OWL uses `OntoRank`, `TermRank`, and the ratio of input concept in the ontology as the ranking measures for the extracted ontologies. `OntoRank` is the ranking measure for ontologies proposed in [4], and `TermRank` is the ranking measure for classes and properties also proposed in [4]. We assume that the more the ontology includes input concepts, the more it relates to the target domain. If two ontologies include the same number of input concepts, the user can select the more popular ontology according to their `OntoRank`. When there are multiple candidates for the input concept due to the ambiguity of the input term, the user can select the more popular concept according to their `TermRank`.

3.5. Constructing Taxonomic Relationships and Other Relationships

In order to support the construction of taxonomic relationships, DODDLE-OWL automatically extracts paths including input concepts from the reference ontologies (acquired existing ontologies for a target domain), merges them, and aligns concepts using ontology alignment. We are developing the ontology alignment function referring to Ontology Alignment Evaluation Initiative [5].

In order to support the construction of other relationships, DODDLE-OWL acquires the properties which define the input concepts or the upper concepts of the input concepts as their `rdfs:domain` and `rdfs:range` from the reference ontologies.

[2] http://www.w3.org/TR/wordnet-rdf/
[3] http://www.w3.org/TR/swbp-skos-core-guide/
[4] http://www.w3.org/TR/rdf-sparql-query/
[5] http://oaei.ontologymatching.org/

Table 1. The Swoogle Web Services, which can be used for domain ontology construction, and their inputs and outputs

Type	Swoogle Web Service	Input	Output
(1)	Search ontology	search keyword	List of `SWO` which relates to the input search keyword
(3)	Search terms	search keyword	List of `SWT` which relates to the input search keyword
(4)	Digest semantic web document	`SWD`	Swoogle Metadata for the input `SWD`
(13)	List documents using term	`SWT`	List of `SWD` defining/referencing/ populating the input `SWT`
(16)	List domain classes of a property	property	List of classes which are used as the `rdfs:domain` of the input property
(17)	List properties of a domain class	class	List of properties which use the input class as their `rdfs:domain`
(18)	List range classes of a property	property	List of classes which are used as the `rdfs:range` of the input property
(19)	List properties of a range class	class	List of properties which use the input class as their `rdfs:range`

4. Implementation

4.1. Acquiring Existing Ontologies using Swoogle

In this Section, we describe the implementation, mentioned in Section 3.2, of acquiring existing ontologies for a target domain using Swoogle. Swoogle provides 19 types of REST web-service interfaces (Swoogle Web Services). When a query URL made by the user is inputted to Swoogle, the user can get the query results in RDF/XML. Swoogle Web Services mainly have `queryType` and `searchString` as their parameters. The `queryType` parameter specifies the type of the web service to call. The `searchString` parameter is given the input search string of the web service. Table 1 shows the Swoogle Web Services available for domain ontology construction, and their input and output. `SWD` (Semantic Web Document) in Table 1 is an RDF document described in RDF/XML, N-Triple, or Notation 3. `SWT` (Semantic Web Term) in Table 1 is an RDF resource with URI being defined, referenced, and populated as classes or properties in `SWD`. `SWO` (Semantic Web Ontology) is a special type of `SWD` which defines many classes and properties.

Table 2 shows the types of Swoogle web services to use and the limiting conditions for each step in acquiring existing ontologies. The `Step` column in Table 2 corresponds to the steps described in Section 3.2. The `Types of Swoogle Web Services to Use` column in Table 2 corresponds to the types in Table 1. In order to reduce the cost of computation time, DODDLE-OWL has limiting conditions for each steps.

4.2. Extracting Elements from Existing Ontologies

In this Section, we describe the implementation, mentioned in Section 3.3, for extracting elements from existing ontologies. To extract elements from existing ontologies, we

Table 2. Types of Swoogle web services to use and limiting conditions for each step in acquiring existing ontologies

Step	Types of Swoogle Web Services to Use	Limiting Conditions
1	(3)	The number of classes and properties for each input term is limited to the top 5 sorted by `TermRank`.
2	(17), (19)	The number of properties which have the classes as their value of rdfs:domain or rdfs:range property is limited to the top 100.
3	(16), (18)	The number of values for rdfs:domain and rdfs:range of each property is limited to the top 100.
4	(1), (4), (13)	The number of ontologies for each input term is limited to the top 10 sorted by `OntoRank`.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX daml03: <http://www.daml.org/2001/03/daml+oil#>
PREFIX daml10: <http://www.w3.org/2001/10/daml+oil#>

SELECT ?label ?description
WHERE {
  {?concept rdfs:label ?label} UNION {?concept rdfs:comment ?description} UNION  {?concept daml03:label ?label} UNION
  {?concept daml03:comment ?description} UNION {?concept daml10:label ?label} UNION  {?concept daml10:comment ?description}
}
```

Figure 4. Extracting labels and descriptions template for RDFS, DAML, and OWL Basic Vocabulary

developed five types of templates for RDFS, DAML, and OWL basic vocabularies. Figure 4 shows the one of these templates described in SPARQL. If DODDLE-OWL executes the `extracting labels and descriptions template` in Figure 4 directly as a SPARQL query, DODDLE-OWL acquires all values of `rdfs:label`, `rdfs:comment`, and etc properties as the SPARQL query result. In order to acquire only the labels and descriptions of an input concept, DODDLE-OWL replaces the `?concept` variable in Figure 4 with the URI of the input concept. In a similar way, DODDLE-OWL replaces the variables in other templates with the appropriate URIs, and executes the replaced templates as the SPARQL query. By building the five types of templates using `?concept`, `?subConcept`, `?class`, `?property`, `?label`, `?description`, `?domain`, and `?range` variables and setting the templates in DODDLE-OWL, extraction of the ontologies' elements described in various scheme is possible with DODDLE-OWL.

5. Experimental Evaluation

5.1. Specification

In this Section, we show the experimental evaluation in a particular field of law to evaluate our proposed method in Section 3 and 4. We evaluated our proposed method especially for the extraction of existing ontologies, and ranking of existing ontologies. The particular field of law used for evaluation is "Contracts for the International Sale

Table 3. The coverage rate of input terms and the number of acquired ontologies, classes, and properties

Coverage Rate of Input Terms	0.717 (33/46)
# Acquired Ontologies	120
# Acquired Classes	331
# Acquired Properties	558

Table 4. The top 10 existing ontologies acquired using Swoogle sorted by ratio of input concept

Rank	# of Input Concepts (OntoRank)	URL of Ontology
1	15 (0.881)	http://www.loa-cnr.it/ontologies/OWN/OWN.owl
2	15 (0.881)	http://www.w3.org/2001/sw/BestPractices/WNET/wnNounsyn_v7.owl
3	12 (642.815)	http://morpheus.cs.umbc.edu/aks1/ontosem.owl
4	12 (0.860)	http://www.cs.umbc.edu/˜aks1/ontosem.owl
5	11 (0.735)	http://rhizomik.net/ontologies/2005/07/FrameNet_1.1_inferred.owl
6	11 (0.726)	http://rhizomik.upf.edu/ontologies/2005/07/FrameNet_1.1.owl
7	11 (0.726)	http://rhizomik.net/ontologies/2005/07/FrameNet_1.1.owl
8	10 (0.875)	http://athena.ics.forth.gr:9090/RDF/VRP/Examples/DCD100.rdf
9	10 (0.875)	http://139.91.183.30:9090/RDF/VRP/Examples/DCD100.rdf
10	9 (1.701)	http://www.cyc.com/2004/06/04/cyc

of Goods"(CISG) [8]. We used the documentation CISG Part-II, which was written in English, as the input document for DODDLE-OWL. A domain expert extracted 46 legal terms as input terms from CISG Part-II. In [3], the domain expert constructed a domain ontology for CISG Part-II with DODDLE-II which is the former version of DODDLE-OWL. In this evaluation, we acquired and ranked existing ontologies for this domain by our proposed method and compared the manually built domain ontology with the results obtained by our proposed method.

5.2. Results

Table 3 shows the coverage rate of input terms and the numbers of acquired ontologies, classes, and properties. DODDLE-OWL used `OntoRank`, `TermRank`, and the ratio of input concept in the ontology as the ranking measures for the ontologies. Table 4 shows the top 10 acquired existing ontologies using Swoogle, sorted by the number of input concepts. Note that these results are as of June 2007.

5.3. Discussions

According to Table 3, the acquired existing ontologies cover approximately 70% (33/46) of the input terms. However, according to Table 4, the highest ratio was approximately only 30% (15/46) when looking at each of the ontologies. For this reason, it is considered that in order to construct a nearly-perfect ontology for a particular domain using the proposed method, DODDLE-OWL would require several reference ontologies.

According to Table 4, most of the high ranked acquired existing ontologies were not domain ontologies but general ontologies described in OWL such as WordNet, Cyc, and etc. However, e-commerce ontologies which included several legal concepts (contract,

offeree, party, payment, proposal, and etc) were ranked 8th and 9th by our proposed method. These ontologies may be reused for this domain. DODDLE-OWL acquires the existing ontologies based on the classes and properties which are acquired in step 1 with a high `TermRank` value. DODDLE-OWL acquires many general ontologies since a number of classes and properties with a high `TermRank` value are included in the general ontologies. DODDLE-OWL also acquired many existing ontologies which defined almost the same classes and properties, but with a different URL. After the user selects the reference ontologies from the acquired existing ontologies, the similar ontologies should be put together before displayed to the user. In this evaluation, DODDLE-OWL only used `owl:imports` property of all the OWL metadata. If the existing ontologies have OWL metadata for version information (`owl:versionInfo`, `owl:priorVersion`, and etc), the ontologies with a version difference can be identified.

6. Conclusion

In order to reduce the cost of building domain ontologies manually, in this paper, we proposed a method and a tool named DODDLE-OWL for domain ontology construction reusing texts and existing ontologies extracted by an ontology search engine: Swoogle. For experimental evaluation, we applied the proposed method to a particular field of law and evaluated the acquired ontologies. As the result, most of the acquired existing ontologies highly ranked by our ranking method were not domain ontologies but general ontologies. However, e-commerce ontologies which included several legal concepts were ranked 8th and 9th by our proposed method. For future works, we would work towards the extension of a function for DODDLE-OWL, constructing taxonomic relationships using ontology alignment from several existing ontologies.

References

[1] Cimiano, P., Völker, J.: Text2Onto - A Framework for Ontology Learning and Data-Driven Change Discovery. In: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems, LNCS 3513. (2005) 227–238

[2] Buitelaar, P., Olejnik, D., Sintek, M.: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In: Proceedings of the 1st European Semantic Web Symposium. (2004) 31–44

[3] Kurematsu, M., Iwade, T., Nakaya, N., Yamaguchi, T.: DODDLE II: A Domain Ontology Development Environment Using a MRD and Text Corpus. IEICE(E) **E87-D**(4) (2004) 908–916

[4] Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P.: Finding and Ranking Knowledge on the Semantic Web. In: Proceedings of the 4th International Semantic Web Conference, LNCS 3729. (2005) 156–170 http://swoogle.umbc.edu/.

[5] Morita, T., Fukuta, N., Izumi, N., Yamaguchi, T.: DODDLE-OWL: Interactive Domain Ontology Development with Open Source Software in Java. IEICE(E) **E91-D**(4) (2008) 945–958 http://doddle-owl.sourceforge.net/.

[6] Buitelaar, P.: OntoSelect: Towards the Integration of an Ontology Library, Ontology Selection and Knowledge Markup. In: Proceedings of the Workshop on Knowledge Markup and Semantic Annotation (SemAnnot2004). (2004) http://olp.dfki.de/ontoselect.

[7] d'Aquin, M., Baldassarre, C., Gridinoc, L., Sabou, M., Angeletou, S., Motta, E.: Watson: Supporting Next Generation Semantic Web Applications. In: Proceedings of WWW/Internet conference 2007. (2007) http://watson.kmi.open.ac.uk/.

[8] Sono, K., Yamate, M.: United Nations Convention on Contracts for the International Sale of Goods. Seirin Shoin (1993) http://cisgw3.law.pace.edu/.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-273

Connecting Distributed Information Using the Semantic Web

Takahiro Abe, Daisuke Wada, Hiromi Kobayashi
Department of Information Media Technology, Tokai University
1117 Kita-kaname, Hiratsuka 259-1292, Japan

Abstract. This paper presents a method of connecting and using information on a distributed network. Web ontology language (OWL) is used to describe and connect information according to the transitive property. Peer to peer (P2P) is used to construct this system. An information management server is used to boost the efficiency for connecting information. Consequently, this system is a hybrid P2P system. A bidirectional depth-first search algorithm is used for seeking an information connection. We develop a prototype system in which reasoning or information association is conducted for ingredients of a dish of cuisine to describe its nutritional characteristics. For this system, JXTA is used as the P2P platform.

1. Introduction

Integration of distributed information has become the subject of many studies because of the development of the internet [1–3, 14–20]. The merits of distributed information systems include not only the distribution of the load in searching but also the capability of continued searching in the event of partial system failure. In pure P2P systems, any computer (peer) can become a server, i.e. an information provider. For that reason, any user can readily become an information provider. An unfortunate drawback of distributed information systems is their inefficiency in information searching. To solve this problem, we propose a hybrid P2P system in which information is distributed in peers called information provider peers (IPP); the management of information is conducted in a server called an information connection peer (ICP). We describe information using OWL in the IPP to connect these distributed information. Information is described using the transitive property and Boolean combinations (union, intersection, complement) in OWL [4–8]. When a user wants to retrieve the relationship of information, the information is connected using these properties of OWL in an ICP in this system. The transitive property in OWL and bidirectional depth-first search algorithm are used to connect information in this ICP. That is to say, knowledge can be produced using some partial knowledge in each peer. Moreover, we implement

this system using JXTA [12] and develop a system in which a transitive search is performed from ingredients of a dish to determine its nutritional value and other characteristics.

2. Our Approach

Each peer is assumed to become an IPP to provide information in this system. The IPPs are distributed in this system. At least one ICP for information management exists in this system. Information is divided using a binominal relationship. As an example, "dish and ingredients," "ingredients and nutrition," and "nutrition and effect" are assumed respectively to be stored in peers. Each binominal relationship can be stored in multiple peers. When a user wants to find a relationship between dish → effect, the connection dish/ingredients → ingredients/nutrition → nutrition/effect are constructed. Thereby, the dish effect is obtained. The merit of such distribution is not only the distribution of the load in searching, but also the capability of degradation in searching in spite of a partial node or link failure of the system. However, the weakness of a pure P2P distributed information system is its inherent inefficiency in search because any search from one peer to some other peers necessitates that all other peers must be searched. To solve this problem, we propose a hybrid P2P system in which information is distributed in each IPP; also, information is managed in an ICP. When a user retrieves the relationship of information, information described using OWL is connected using an ICP in our system. A dictionary peer is assumed to be provided to avoid inconsistency of the description of the class name and the relationship of classes. In this paper, we do not intend to discuss the method of construction of a dictionary peer further. Although we assume that an actual system has more information, we present a model comprising 'dish and ingredients," "ingredients and nutrition," and "nutrition and effect" for ease of understanding. As an example, information related to a dish and ingredients is accompanied with recipe and other information. However, description of all information using OWL seems to be difficult. Therefore, to give an example, only the relationship between a type of cuisine and its ingredients is described using OWL in this system. Other accompanying information is omitted for simplification. The relationship of classes that are partial knowledge are stored in distributed IPP written using OWL. These partial information classes are connected in an ICP. Reasoning paths are formed using transitive properties between classes that are accumulated from multiple IPPs in the ICP. Consequently, when a user wants to retrieve an instance in class D related to instance a_1 in class A, this user first queries an ICP. Then ICP returns Peer IDs (PIDs) which belong to the IPP on paths to this user. The paths are formed based on reasoning paths from classes A to D in ICP. Subsequently, the user queries each IPP based on these PIDs to obtain related instances. Finally, the user can obtain an instance (or instances) in class D.

We develop a prototype system using JXTA, which is open source software distributed by Sun Microsystems as a P2P platform.Each peer is identified using PID in JXTA. Each peer activates in an independent and asynchronous manner to other peers. An advertisement consisting of an XML document is used to notify other peers of services in JXTA. We also use Jena [9–10], which is open source

software distributed by the HP Laboratory to model and retrieve information using RDF Data Query Language (RDQL) [11] ontology.

3. Procedure

The system procedure is divided into a path connection procedure and information retrieval procedure. The path connection procedure occurs periodically in an ICP. An IPP sends PIDs and class information to ICP to join this system. The connection of classes is performed using the transitive property in OWL in ICP. The path connection procedure consists of a reasoning procedure using transitive property and a bidirectional search algorithm based on the depth-first search algorithm in ICP.Reasoning paths are formed using class information and PIDs gathered from the IPP. Class information is described using a binominal relationship as dish $\rightarrow$ ingredients. Using the transitive property, A $\rightarrow$ B and B $\rightarrow$ C, a path A $\rightarrow$ C is obtainable. Connection paths are formed using reasoning paths based on the PIDs of IPP. The bidirectional search algorithm, which consists of a binary search algorithm and a depth-first search algorithm, is used to raise the efficiency of this procedure. This procedure is presented in Figure 1 and the core behavior is illustrated in Figure 2.

The information retrieval procedure occurs when a user retrieves distributed information. This user first queries to the ICP when the user wants to retrieve an instance. The ICP returns the PID which belongs to the IPP on paths to this user. Next, the user queries each IPP in sequence based on this PID. Subsequently, retrieval is conducted using RDQL in IPP. The IPP automatically returns these results to the user sequentially. Finally, the user can obtain an instance of the goal. This procedure is shown in Figure 3 and portrayed in Figure 4. Procedure (1) to (4) are performed automatically in Figure 4.

We next estimate the computational complexity of the algorithm described above. The average order of the mixture of the binary search algorithm and depth-first algorithm is of $O(m \cdot n \log n)$ where n is the number of PIDs and m is the number of groups with the same PID, i.e. the number of IPPs.

We developed a prototype system that is useful to retrieve nutritional information about a dish from its ingredients. We used a simple model composed of "dish and ingredients," "ingredients and nutrition," and "nutrition and effect" for obtaining cooking information. Using this system, we can obtain information in answer to a query such as "What is the physiological effect of eating curry?" —Answer: "It improves immunity."

4. Related Work

Related work that combines Semantic Web with P2P technology is as follows. Some work [1, 16–17] have been aimed at discovering semantic mapping across hierarchical classification using P2P. The same vocabulary in different hierarchical structure are discoverd using CTXMATCH algorithm in [17]. Oyster [18] is a system for exchanging ontology metadata among communities in order to build an application faster in P2P. A user can import ontology files and extract the ontology metadata in this system. Semantic P2P search systems are developed for the distribution of the load in searching [12, 19].

```
variables:
CC: Current Class
CC.fwd: Current Class in forward search
CC.bwd: Current Class in backward search
NC: Next Class;  BC: Before Class
FC: First Class;  LC: Last Class
CC_PID: Current Class PeerID
CC_NSPID: No Search PeerID at Current Class
CC_PID.fwd: Current Class PeerID in forward search
CC_PID.bwd: Current Class PeerID in backward search
NC_PID.middle: PeerID middle within Next Class
NC_PID.low: PeerID low within Next Class
NC_PID.high: PeerID high within Next Class
procedure:
BS (): // Binary Search
while(CC_PID ≠ NC_PID.middle)
 NC_PID.middle ← (NC_PID.low + NC_PID.high) /2;
 if(CC_PID < NC_PID.middle)
   NC_PID.high ← NC_PID.middle - 1;
 else(CC_PID > NC_PID.middle)
   NC_PID.low ← NC_PID.middle + 1;
Main ():
CC.fwd ← FC;  CC.bwd ← LC;
call upon BS(CC_PID);
while(CC_PID ≠ NULL)
 if(CC_PID.fwd = CC_PID.bwd)
   call upon BS(CC_NSPID);
 else
    CC ← NC;
    call upon BS(CC_PID);
while(BC ≠ NULL)
 CC ← BC;
 while(CC_NSPID ≠ NULL)
   call upon BS(CC_NSPID);
END
```

Figure 1 Bidirectional search algorithm

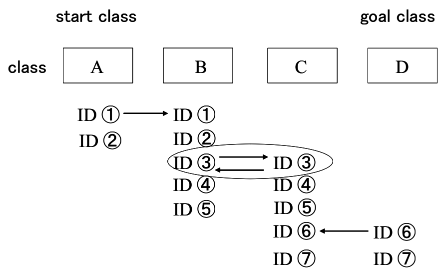


Figure 2 Bidirectional search

@User make Query_search_info (PeerGroupID)
send Query_search_info to ICP

@ICP search PID using Query_search_info
send PID to User

@User send Query_search_info to IPP using PID

@IPP search Distributed_info using Query_search_info
send Distributed_info to User

@User make Query_search_info using Distributed_info
send Query_search_info to IPP using PID

Figure 3 Information retrieving procedure

As for inference, Semantic Web Rules Language (SWRL) is proposed [13]. However, it seems that application systems using SWRL have not been advanced.

Besides JXTA, Tryllian [14] and JADE [15] are well-known agent platforms for developing a P2P system. However, JXTA is one of the most freely expandable platform so far. Recently, JXTA has been used for discovering Web services in [20]. We have developed a system that combines OWL with JXTA to obtain knowledge from some partial knowledge in each peer. Inference is conducted based on the binominal transitive property in OWL.

5. Concluding Remarks

This paper presented a hybrid P2P system consisting of information provider peers (IPP) and an information connection peer (ICP). The objective of this system is gathering of distributed partial knowledge and discovery of a new relationship of partial knowledge. All peers can become IPPs. In fact, an ICP is needed for efficiency of retrieval. A dictionary peer is necessary to unify the description of ontology. However, the construction of the dictionary peer is not addressed in this research. Additional research about the dictionary peer is necessary for further development.

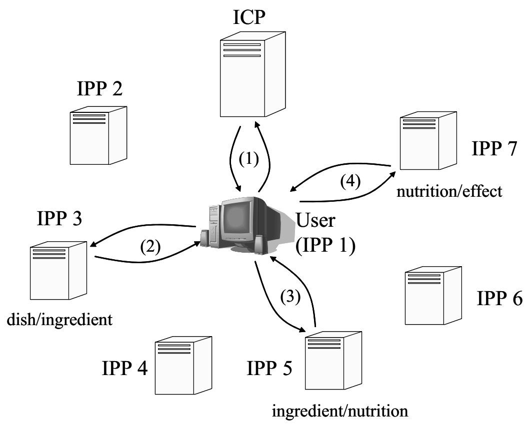


Figure 4 Information retrieving

References

[1] H. Stuckenschmidt and F.v. Harmelen,*Information Sharing on the Web: 2nd Edition*, Springer, 2008.
[2] H.P. Alesso and C.F. Smith, *Developing Semantic Web Services*, A K Peters LTD., 2004.
[3] N. Eisinger and J. Maluszynski (eds.),*Reasoning Web*, Springer, 2005.
[4] G. Antonius and F.v. Harmelen,*A Semantic Web Primer*, MIT Press, 2004.
[5] K.K. Breitman, M.A. Casanova and W. Truszkowski, *Semantic Web*, Springer, 2007.
[6] N. Zhong, J. Liu and Y. Yao,*Web Intelligence*, Springer, 2003.
[7] A. Gomez-Perez, M. Fernandez-Lopez and O. Corcho, *Ontological Engineering*, Springer, 2004.
[8] R. Mizoguchi,*Ontology Engineering*(in Japanese), Ohmsha, 2005.
[9] Jena Homepage: http://jena.sourceforge.net/
[10] IBM developer works: http://www.ibm.com/developerworks/jp/java/library/j-jena/
[11] RDQL Homepage: http://www/w3/org/Submission/2004/SUBM-RDQL-20040109/
[12] B.J. Wilson, *JXTA*, New Riders Publishing, 2002.
[13] SWRL Homepage: http://www.w3.org/Submission/SWRL/
[14] K. Burbeck, D. Garpe and S. Nadjm-Tehrani, Scale-up and Performance Studies of Three Agent Platforms, *International Conference on Performance, Computing and Communications (IPCCC04)*, pp.857-863, 2004.
[15] F. Bellifemine, A. Poggi and G. Rimassa, JADE: A FIPA2000 compliant agent development environment, *Internatinal Conference on Autonomous Agents 2001 (Agents 2001)*, pp.216-217, 2001.
[16] M. Bonifacio, P. Bouquet, G. Mameli and M. Nori, Kex: A Peer-to-Peer Solution for Distributed Knowledge Management, *4th International Conference on Practical Aspects of Knowledge Management (PAKM2002)*, 2002.
[17] P. Bouquet, L. Serafini and S. Zanobini, Peer-to-Peer Semantic Coordination, *Semantic Web Applications and Perspective Workshop (SWAP2004)*, 2004.
[18] R. Palma and P. Haase, Oyster-Sharing and Re-using Ontologies in a Peer-to-Peer Community, *International Semantic Web Conference (ISWC 2005), LNCS 3729*, pp.1059-1062, 2005.
[19] K. Nakauchi, Y. Ishikawa, H. Morikawa and T. Aoyama, Exploiting Semantics in Unstructured Peer-to-Peer Networks, *IEICE Trans. Communications*, E87-B(7), pp.1806-1817, 2004.
[20] M Pantazoglou, A. Tsalgatidou and G. Athanasopoulos, Discovering Web Services and JXTA Peer-to-Peer Services in a Unified Manner, *International Conference on Service-Oriented Computing (ICSOC 2006), LNCS 4294*, pp.104-115, 2006.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-278

Exchanging the Business Concepts through Ontology System

Toshihiko Tsumaki [a]
[a] *Nihon Unisys Ltd.*

Abstract. The businesses reorganization by the merging and the dividing is progressing in order to strengthen of their competitiveness in the market. In the business reorganization based on cooperation the businesses activities, it is necessary to understand the terms and the concept each other. We propose a new approach that uses the interaction instead of the integration for supporting the cooperation between different business units. The concepts in the business unit are independently expressed by means of the ontology technique and implemented into the corresponding ontology units.

Keywords. Ontology, Business unit, Application component, Conceptual model

Introduction

The popularization of information system is also progressing by new information technology: internet, personal computer, GUI, and so on. These new movements are causing new problems to the development of software. The big problems are how to cope with the increasing variety and the growing volatility of the business environments. The other hands, the business structure becomes complicated and is rapidly changing. The business competition in the market increase violent by the business globalization and the economic relaxation, and then, many industrial companies are trying to advance the businesses reorganization by the merging and the dividing to strengthen of their competitiveness in the market. The variety and the volatility of software increase further by the business reorganization. Many of business reorganization are based on cooperation and not integration of the businesses activities because the integration deprives the other enterprise of independency. We propose an approach that uses the interaction instead of the integration for supporting the cooperation between different business units.

1. Conditions of Ontology System

Each business unit in a connected type business is supported by individual software component and the components have to be cooperated with each other for supporting the business operation. In the business, it is necessary for each component to understand the business concepts of the partners when cooperation is required. Therefore, to hold the partner's concept in the component itself is not only useless but also make difficult the system building and maintenance. It is necessary to separate the definition of concepts from the components to support business operation. This is an ontology system. The ontology system is composed of ontology units for each business unit. There are some problems for defining and converting the business concepts on the ontology system. We examine about following four problems in this paper: Expression

of a concept, Accuracy of concept conversion, Coping with concept change and Guaranty of ontology's correctness.

1.1. Expression of Business Concept

In order to deal with business concepts, it is necessary to define a common method to express it.

- **Extension**: An extension is an extent of things that the concept is applied. These things are also concept. We express a relation between the original concept and the each extended concept by 'a-kind–of' relationship.
- **Intension**: An intension is a whole of common essential attributes to all of the extended things. We express the relation between the concept and the each attribute by 'has-a' relationship. If the concept has indispensable intensions, it is expressed by 'must- has-a' relationship.
- **Example**: An example is a concrete thing that has the intension of the concept. The concept may be often a role to be fulfilled by the thing. We express the relation between the concept and the each example by 'is-a' relationship.
- **Collaborator**: A collaborator is also a concept that associates with and is cooperatively active with the concept. We express the relation between the concept and the each collaborator by 'associate-with' relationship.
- **Level**: The concept is distinguished by not only a kind of the attributes but also a degree of the attributes: size, force, etc. a level is a qualitative criterion of attributes. We express the relation between the attribute and the level by 'degree-of' relationship.

ConceptNet has a lot of relationships for representing the relations between concepts in common sense: Is a, Has property, Location of, Used for, and so on [1]. Also, OWL to exchange the information on Web [2] has subject, object and predicate. Although strong expressing capability of these tools copes with the open domain, it is not required in our closed business domain. We used Class diagram in UML to model the concept structure Figure 1 is an expression of the concept by Class diagram.

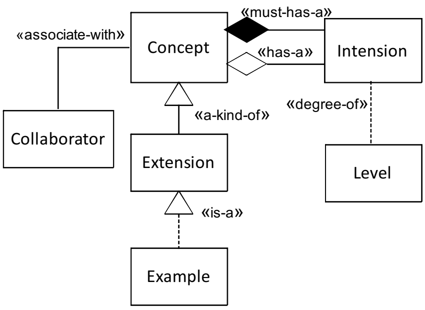


Figure 1. Concept model by Class diagram

1.2. Conversion of Concept

There are two kinds of the interaction between collaborated business units. One is one way interaction such as a notification of information and another is two way interactions such as query - answer and request -response. When both business units

use same name to one concept, it is no problem. However, if different business units use different name for one concept, the interaction will failed.

Another case of interaction fails is caused of disagreements in the name and the meaning of the concept. It might be possible to discover the fails through additional information such as the extension or the interaction context. However, it is an accidental phenomenon and cannot be always guaranteed. In the real world, relation between the name and meaning of concept is ambiguous though duplication of term is not permitted in an information system. The problem is caused in the polysemous word or the synonym when the different business units try to communicate.

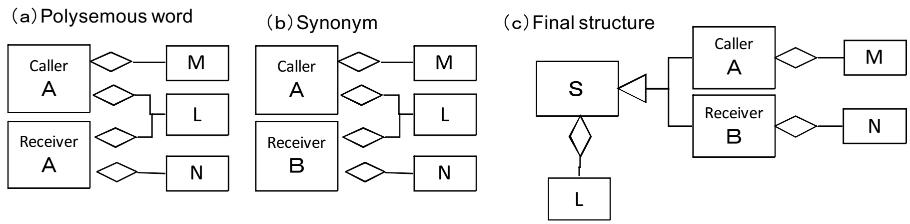


Figure 2. Polysemous word and Synonym

Figure 2 shows (a) polysemous word and (b) synonym. The interaction between the business units is required to make the connection succeed because the concepts used by each business unit are defined in the corresponding ontology unit. The interaction is defined as a concept exchange. A and B are name of the concepts and L, M and N are name of attributes. (a) and (b) can be converted to (c).

1.3. Meaning Conversion of Concept

In this paper, we considered that a concept is composed of a name and some attributes. However, the concept in the real world has various attributes and is ambiguous. Therefore, it is difficult to judge whether a concept is same or different [3]. The concepts are identified, and divided by the difference of the perspective. We use two indicators that are a similarity and an identity in order to judge whether the concepts are same or different.

- **Similarity**: a difference of the level of the attribute.
- **Identity**: a difference of the kinds of the attributes.

First is the similarity judgment about a level of value in order to decide whether a target attribute of the concepts between different business units is same or not. Evaluation criteria for the attributes are different according to the business unit but a common criterion is required in the interaction of business units. If a standard criterion is established and opened to the public in the business world, it is useful to use it. If there is no standard criterion, it is necessary to adjust the criteria based on previous result data between the business units.

Next is the identity judgment about the kinds of attributes in order to decide whether a target concept between different business units is same or not, if it is recognized that the each attribute is same. There are six relation patterns: Complete agreement, Partial agreement, Inclusion, Share, Division and Disagreement. When the concepts are decided to be same one, it is necessary to make the name agree. If these concepts are given different names in the different business units, the names have to be agreed during the interaction of the ontology units.

1.4. Coping with Concept Change

Software is often changed by the change of business rules, the introducing of new information technologies and the growth of users. So these causes are unavoidable phenomena for software, that various ideas have been proposed in software engineering:

- Minimizing of the influence of change by the modulation of software.
- Saving of the labor to change by the configuration management of software.
- Building of the unchangeable software by anticipation of the future business.

The change of concepts may cause the problems of interaction between business units supported by the ontology units. Figure 3 shows the concepts conversion forms between the ontology units. Little black circles indicate the conversion points. The mediator holds an own ontology. We introduced (c) because of the frequency of the concept conversion and the changes of the contents.

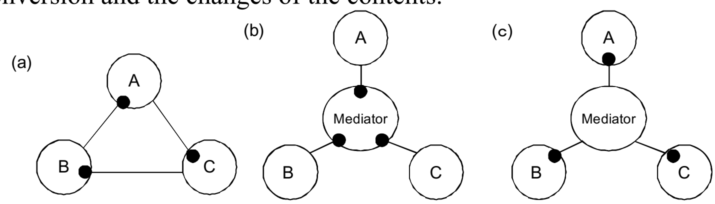


Figure 3. Methods of interaction of ontology units

1.5. Guaranty of Ontology's Correctness

Concept in each ontology unit has to be correct because it is used by the foreign ontology units. The correction of ontology units has two aspects. One is a formal inconsistency that means the conceptual system has no conflict and is guaranteed by means of the logical implementation languages such as Prolog. Another is a meaning validity that means the concept of each business units is exactly expressed. In this section, we examine the meaning correctness of the ontology units. The main means to obtain the proper model is still to introduce the expression methods for enabling the entire intuitive understanding and the review by participants with different perspectives. We establish two criteria for judging the validity of meaning.

- **Exactness**: There is no gap between the concept in the real world and model.
- **Accuracy**: The examples can satisfy the capability defined by the attributes in the model

When there is more than one perspective, it is possible to divide the business unit. The application software components and ontology units are developed by each business unit. If there is an application software component for supporting more than one business units with different perspectives, it is necessary to integrate the perspectives.

2. Structure and Mechanism for Ontology System

In this paper, we took up three problems: the concept exchange, the concept change and the concept correctness in the ontology system.

The ontology system composed of the ontology units corresponding to business units. This structure is not easily influenced by the change of other business units. Figure 4 is a conceptual structure of the business units and the ontology units.

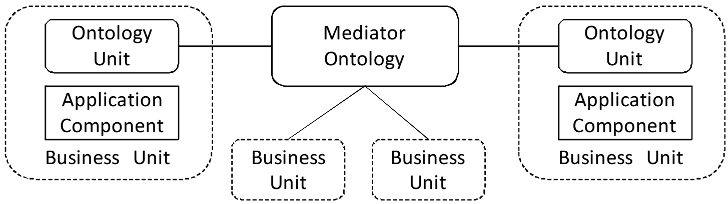


Figure 4. Structure of the ontology system

2.1. Ontology Units

Our ontology system is composed of two types of ontology: **ontology unit** and **mediator ontology**. The ontology units represent the attributes and relations about the concepts of the business units. Each business unit has unique perspective in the domain. A business unit may have more than one contradicting viewpoint: managements and worker, customers and suppliers, designers and implementers and so on. But they are usually integrated to one viewpoint. If it is difficult to integrate them, the business unit should be divided. However, many of attributes in a same concept are common even if they are in the different viewpoints. So, it is possible to express them by the inheritance of common attributes. The 'a-kind-of' and 'is-a' structure is efficient to express the common attributes of entities in different ontology units [4]. Some studies define a subset of attributes as a role. This idea introduces a layer structure of ontology units. The roles are common attributes of a certain concept. An abstract role can be created by a set of common attributes and the correction of abstract roles forms an abstract ontology unit. We are going to use a conceptual model to keep the consistency.

2.2. Mediators

Mediator is a mechanism for supporting the interaction between components. Mediator is also one of the ontology units because it has the concepts that are composed of rules and knowledge to fill the gap between ontology units. The mediator has two important functions: **Translation of concept** and **Judgment of the tolerance level**.

References

[1] Liu, H., Singh, P. ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal* **22** (2004).

[2] Deborah L. McGuinness and Frank van Harmelen, *Web Ontology Language (OWL): Overview*, 2004-02-10, W3C Recommendation. <http://www.w3.org/TR/owl-features/>

[3] Gal, A., Modica, G., Jamil, H., Eyal, A. Automatic ontology matching using application semantics. *AI Magazine archive*, **26**(1) , (2005), 21 – 31.

[4] Brachman, R. J. What IS-A is and isn't: an analysis of taxonomic links in Semantic networks, *IEEE Computer* **16**(10), (1983), 30-36.

Internet-Based Interactive Applications

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-285

Holistic Electronic & Mobile Government Platform Architecture

Athanasios Karantjias[1] , Spyridon Papastergiou[1], Despina Polemi[1]

[1] University of Piraeus, Karaoli & Dimitriou 80, Piraeus 18534, Greece
{karant, paps, dpolemi}@unipi.gr

Abstract. The growth of high speed e/m-services imposes requirements including: *security*, which fosters user trust in mobile services, *interoperability*, which enables cross-domain mobile service communication and wireless application integration; *high administration* on the services, organizing embedded logic of applications into separate and easily changed "state machines" to allow new level of processes within governmental services, and *scalability* and *extensibility*, which ensure that existing services and modules are not degraded and can be easily extended in new, more advance ones. This paper proposes a secure and interoperable governmental platform architecture for complex e/m services, which integrate peak XML-based technologies.

Keywords: High-administration, Extensibility, Interoperability, XML Security, Scalability, e/m-Government

1 Introduction

The impressing penetration rates of *electronic and mobile (e/m)* networks provide the unique opportunity for all countries to use e/m-services and accelerate their entrance in the digital society, strengthening the fundamental structure of *Governmental Organizations (GOs)* and enhancing the collaboration of the public sector [1]. Consequently, nowadays the need for building real interoperable, secure, high administrative and easily deployable e/m governmental frameworks and platforms is imperative.

Centralized e-government approaches, such as Web portals [2] didn't succeed to address these needs, since their major focus was built upon content distribution, embedding the business logic of all services directly into enterprise applications.

Extensively used distributed e-government solutions, such as the *LGOL-Net*, *LGOL-X*, *LGOL-Flow* and *Aplaws* enterprise systems, deployed under the LAWs national project in UK [3], the *MMBase* relational object mapping tool [4] used in Netherlands and Germany, the *Mioga 2 groupware solution* [5] in France, aimed to develop a set of assets that can be implemented on a modular basis and depended on local governmental technological and information management maturity. In addition to these, many EU research and development projects on e-Government, such as eMayor [6], aimed to provide secure, interoperable and affordable Web Services platforms for GOs across Europe.

However, the lack of standardised, open source technologies in building advance content repositories, middleware messaging and customizable content management systems, as well as business process management, strong security and interoperable mechanisms and modules, at the time that these solutions were deployed, necessitates the integration of more advanced ones. Nowadays the open-source world appears mature in providing stable technology premises to build real interoperable and secure enterprise systems.

Moreover, the facts that the mobile aspect is partially covered, and that many technology frameworks and tools used for their integration are not supported in our days, imposes their replacement with more synchronous ones. These must address all mobile requirements for service and technology interoperability, strong security, user-friendliness and low cost, overcoming bandwidth and performance considerations, as well as existing development difficulties. In addition they shall be appropriately designed and consisted from independent modules that will be able to provide high administrative e/m-services with one, single implementation, differentiating only the end user layer.

The adoption of peak technologies based on XML as the mean to achieve interoperability, openness and high-administration is a well promising approach in both the electronic and mobile environments. Their functionality is delivered over world-wide accepted protocols and standards, combined with powerful security technologies such as *Public Key Infrastructure (PKI)* [7] and XML Cryptography [8].

This article presents a secure, interoperable platform architecture, which operates in a synchronous e/m-governmental framework, offering high-administrative e/m services. Section 2 analyses the core design principles of the proposed platform, Section presents the architecture and the core enterprise services of the platform and Section 4 draws Conclusions.

2 Core Design Principles

The fundamental benefit of the presented platform is the consistent usage of a standards-based architecture, which integrates easily modified and expanded functionality, and re-engineered e/m-services. It's design and implementation core principles, ensuring that each citizen/organization perceives the quality of the services provided and trusts these that being delivered, are analyzed in the following paragraphs.

Interoperability

Interconnecting many disparate *Governmental Organizations* (GOs) is a difficult task, requiring easily identifiable and publishable e/m-services as well as interfaces for the establishment of secure and reliable connection points [9]. The adoption of Web Services and advanced XML-based technologies surely promotes the concept of atomic, self-contained, services, which are accessible and available to a multitude of applications in different and demanding environments, offering the promise and hope of integrating these applications in a seamless fashion [10].

However, large enterprise frameworks are built upon different technologies and platforms, and their integration across businesses has never been a trivial task. The emergence of the *Business Process Execution Language (BPEL)* [11] and its embracing by major vendors and IDE tools [12][13] provide a higher-level description language to specify the behavior of Web Services' framework, orchestrating them into business processes. It actually defines an interoperable model that facilitates the expansion of automated process integration in both the intra-corporate and the business-to-business space [14].

Security & Trust

Security and Trust are key enablers of future digital governmental environments, based on many interacting objects, devices and systems. Although Web Services permit enterprises to create interoperable services-based applications, their original definition didn't include a built-in security model 14. The use of the *Secure Socket Layer (SSL)* 15 protocol to protect communications among service endpoints does not provide the granularity and flexibility, required for more advanced Web Services scenarios, where the endpoints might not have a direct channel to each other.

Second generation PKIs and advanced XML cryptography mechanisms support a large scale deployment of a number of security services, such as origin authentication, content integrity and confidentiality, and non-repudiation, establishing trust chains at local, national and international level.

OASIS *WS-Security (WS-S)* [17] defines a core specification for securing SOAP messages, plus several extensions for integrating user or service identity information within these, based on XML cryptography. W3C *XML-Digital Signature* (*XML-DSIG*) [18] standard defines the appropriate way for rendering digital signatures in XML, making them human-readable, easily parsed, platform independent, and generally more advantageous for workflow environments than preceding standards like the *Public Key Cryptographic Standard #7 (PKCS#7)* [19]. Correspondingly, W3C XML Encryption [20] standard allows the selective encryption of arbitrary portions of XML documents, allowing seamless integration into workflow processes.

In addition to these, the use of a brokered authentication system, which issues XML-based security tokens, integrating the *Web Services Trust (WS-Trust)* [21] standard, enables user authentication in heterogeneous environments providing authorization and auditing advanced mechanisms [22].

Scalability & Extensibility

Nowadays, e/m government enterprise solutions due to their increased usage in everyday life, need to be simple, open and reconfigurable, providing easily reengineered services and taking into account that the large number of citizens needs also to be served with acceptable quality of service levels. Consequently, these frameworks demand the creation of a dependency between business and information technologies in order for GOs to maintain scalable and extensible systems that efficiently support business activities.

The adoption of *Service Oriented Architectures (SOA)* is the solution for moving away from single-platform closed systems to a set of loosely coupled services that offers all the benefits of a monolithic application, without its constrains [23].

The use of *Business Process Management (BPM)* frameworks for modeling all business processes, organizes the embedded logic of an application into separate and easily changed "state machines". This allows new levels of processes to be defined within businesses, fulfilling the compartmentalized needs of different GOs departments and providing scalability and extensibility with minimum effort. Therefore minimization of costs can be achieved for every future improvement, which is very important for every governmental organization that lacks of financial resources and personnel (ex. Small and Medium GOs).

User-Friendliness & Accessibility

Synchronous governmental systems integrate complex mechanisms and heavy enterprise operations, which need to be transparent to all end users (citizens, organizations and civil servants). Automated processing is linked to the operational costs of the e/m-services for public organizations, which satisfy a great number of requests with the need of only a handful of personnel under normal operations. This could be leveraged by integrating e/m-based technologies, which give to citizens and businesses the opportunity to perform part of processes by themselves, using Web Services for reducing human intervention in *Business-to-Citizen (B2C)* and *Business-to-Business (B2B)* transactions to the largest possible extent.

The use of strong, open source frameworks and rich *User Interface (UI)* technologies such as *Spring Web Flow (SWF)* [24], *Java Server Faces (JSF)* [25], *RichFaces* [26] and *Facelets* [27], provide a user friendly web component based architecture, improving system performance. Moreover, they offer to developers the opportunity to easily build interactive interfaces with basic form controls and efficient, reusable operations, while architecturally they are able to separate the presentation logic (the "what") from the UI component's business logic (the "why" and "how").

3 e/m-Government Platform

The synchronous e/m Government platform uses SOA design principles throughout, consisting of the following tiers:

- The *Client Tier,* which integrates all the necessary components for e/m accessing all external entities and requesting advanced governmental services.
- The *Interaction Tier*, which comprises all required interfaces, which establish communication channels with every e/m entity outside the platform.
- The *Main Enterprise Tier*, which contains the core platform components and integrates basic but synchronous services.
- The *Secondary Enterprise Tier*, which manages the choreography of the core platform services, implementing their business logic.
- The *Integration Tier*, which comprises the required adaptation components that sit "on-top" of many existing/ legacy governmental systems.

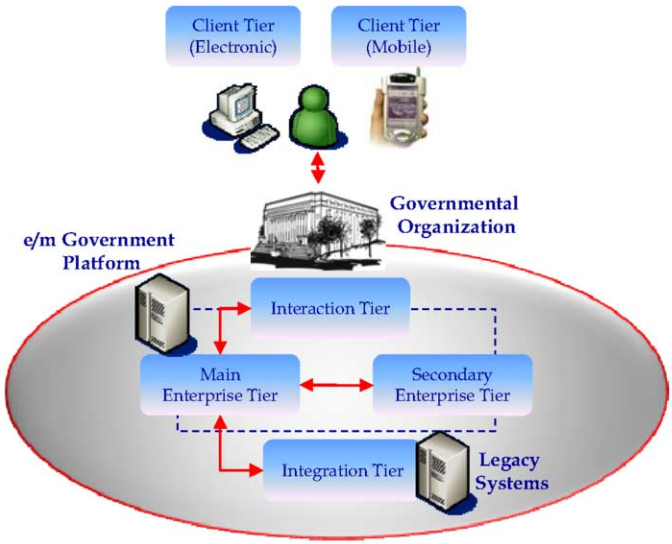


Fig. 1. Tiered Architecture of e/m Governmental Platform

Each one of these tiers, which are analyzed in the following paragraphs, contains either independent components or *Basic Engineering Objects (BEOs)* [28], communicating with each other through clearly defined operation channels.

3.1 Client Tier

The client tier consists of τhe electronic node, which provides access to all entities via the use of browsers and smart-card readers, and the mobile node, which consists of a mobile application and its advanced modules, operating in a mobile device.

Electronic Node

The electronic node consists of a capsule that contains the Smart-Card Reader and another that includes the components, which operate inside the browser environment. The processing of requests and display of e-documents is carried out by the end-user's browser, which uses a security plug-in, installed with the adopted smart-card reader, or a specific ActiveX component, in order to access all security credentials stored in smart-cards, and the *Document Object Model (DOM)* [29] of the browser.
The operation channel is connected with the *Interaction Tier*, which operates in the main platform.

Mobile Node

The mobile Node comprises of four core capsules, as depicted in Figure 2, the functionality of which should be included in every synchronous mobile implementation that intend to be used in future governmental frameworks.

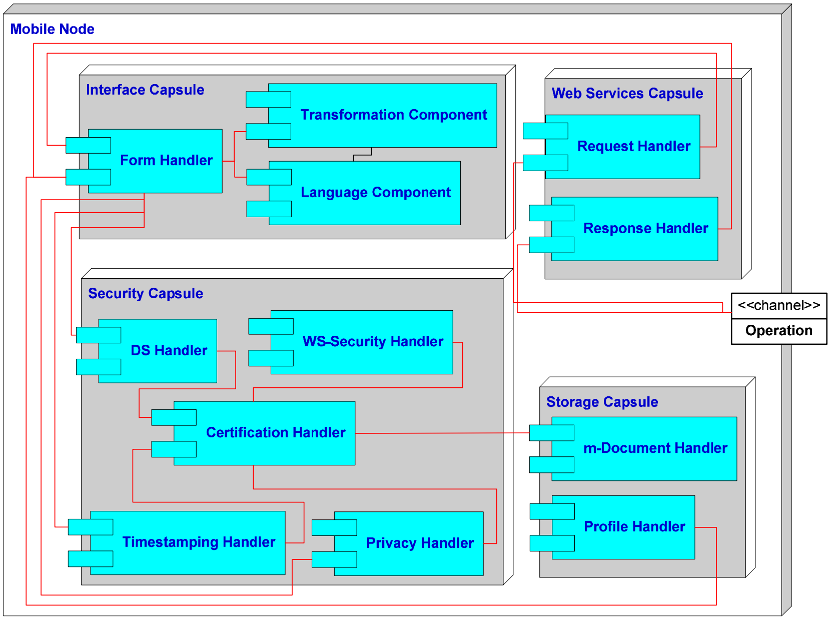


Fig. 2. Mobile Node - Client Tier Analysis

The *Web Services Capsule* implements all formulation and handling mechanisms of the transmitted messages to external communicating entities (*Request Handler*). It actually encloses all clusters of data into Web Services, integrating as well the reception and extraction of the main body mechanisms (*Response Handler*) on every other end of communication.

The *Interface Capsule* implements the transition between m-forms (*Form Handler*) during the process of user interaction, the selection and automated adjustment of language and character set on these forms (*Language Component*), and the transformation of the given data into a format compliant with the adopted m-government XML schemas (*Transformation Component*).

The *Security Capsule* integrates strong security mechanisms on the mobile device. It creates and verifies XML digital signatures on the m-documents (*DS-Handler*), which are automated structured from the m-forms, as well as the hash values of the signed m-documents and requests for valid timestamps (*Timestamp Handler*) from the TSA. All SOAP messages are digitally signed and encrypted (*WS-Security Handler*) using strong cryptographic credentials which are stored and handled in this capsule (*Certification Handler*). Furthermore, it creates the appropriate requests for obtaining valid authorization tokens from the brokered Authentication server (*Privacy Handler*), receives and handles them by automatically embedding them in messages to be sent to the main e/m government platform.

Finally, the *Storage Capsule* stores and handles the created and received m-documents (*m-Document Handler*), and the various profiles of users (*Profile Handler*) on the mobile device. Depending on the authenticated user, many fields (ex. name, surname, V.A.T. number, etc), of the m-forms are automatically fulfilled.

3.2 Interaction Tier

The Interaction Tier communicates with all end-users (citizens, organizations and civil servants) through stream channels, and the enterprise tier through two operation channels.

It contains the main servlets, which process all e/m requests via HTML and Web Services correspondingly, and ensures that each one of these is routed to the proper component. Specifically, the *Request Handler Module* initially receives all e-requests and processes them appropriately.

The *Form* and the *Web Content Pages* modules integrate and handle all UI components, technologies and frameworks for e-access on the platform. These modules integrate the SWF technology, which allows the use of any object as a command or form object, without duplicating business properties. It actually manages all web application page flows, integrating with existing frameworks (ex. JSF) in both servlet and portlet based environments.

Especially for governmental business processes, which benefit from conversational model as opposed to a purely request model, SWF is a great solution, allowing the capture of logical page flows as self-contained modules and guiding users through controlled navigations that drive to actual business processes.

The *SOAP Proxy* module handles all m-requests based on Web Services, while the *Security Module* is responsible for integrating the WS-S technology, encrypting/decrypting SOAP messages, creating/validating XML digital signatures on them, and embedding/exporting the SAML authorization tokens, provided from the STS server.

3.3 Main Enterprise Tier

The main Enterprise Tier provides the basic functions used as a whole to perform a number of primitive services.

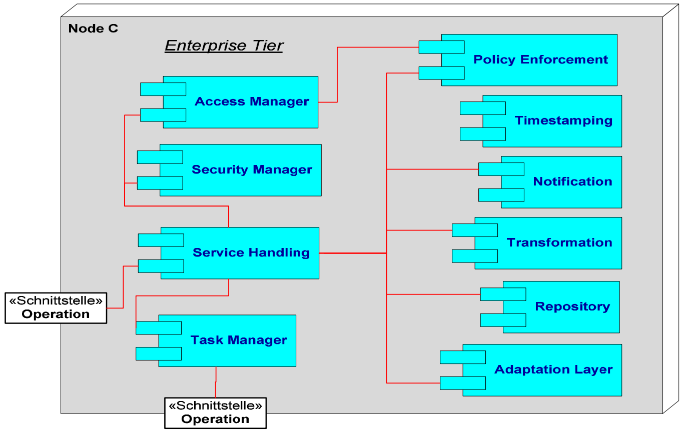


Fig. 3. Main Enterprise Tier Analysis

As depicted in Figure 3, *Service Handling* is the core component that receives request messages from the interaction tier and the secondary enterprise tier, responding appropriately, while cooperating with the *Notification* component, which handles all notification messages to end users. It extracts the authorization token and forwards it to the *Access Manages* component, which handles all access control requests and validates these SAML assertions. Additionally, it enforces predefined policies, handled from the *Policy Enforcement* component, integrating the *OASIS eXtensible Access Control Markup Language (XACML)* [30].

Security Manager corresponds to the core security module being responsible for handling all security credentials and implementing mechanisms, such as the creation and validation of simple (XML-DSIG) and long term validity (*XML Advanced Electronic Signatures - XAdES*) [31] e-signatures, since signed public documents may have to bear signatures lasting for a long time. Therefore it cooperates with the *Timestamping* component, through the *Service Handling*, for obtaining timestamp tokens from the TSA entity.

This tier also includes the *Repository* component for the input and output of e-documents to/from a given registry in the GO, the *Adaptation Layer* component where all required Web Services, which communicate with end-points and interfaces at legacy systems, are handled, and the *Transformation* component, which actually handles all transformation mechanisms of data from one form to another, providing the appropriate input to all other components in the platform.

3.4 Secondary Enterprise Tier

Taking into account that traditionally, process modeling has relied on proprietary tools and methodologies, raising the cost and limiting shared understanding, the secondary Enterprise Tier implements powerful new process modeling standards, adding advanced features in managing the business logic of the enterprise services. It defines and mixes human based actions, creating business rules based on workflow data. This tier integrates all governmental business services, making use of all advanced services, presented in the previous described tiers.

Java Business Process Management's (jBPM) [32] graphical notation is used from both, technical and business professionals in defining the actual governmental business processes, which are implemented with the help of an executable web service orchestration language such as the BPEL.

3.5 Integration Tier

The Integration tier may consist of several backend systems, which reside and operate on the GOs providing external services and data. Appropriate interfaces, based on Web Service, are required in order for them to be accessed from the proposed platform and specifically from the Adaptation tier component which resides and operates in the main enterprise tier.

4 Conclusions

The main objective of this paper is to present a secure, interoperable e/m-Government platform which enables GOs to make the next step on digital society. This platform addresses core e/m-Government requirements such as security, interoperability, transparency, scalability and high administration.

It's "out of the box functionality" enables the creation and provision of advanced government business services with a simple installation and customization, enabling:

- Developers to easily extend its functionality reusing several architecture modules when implementing new governmental services.
- Business analysts to reengineer the deployed enterprise services, using graphical interfaces of the BPM system on runtime environment, and redeploy the service.

Its adoption enables faster interaction among entities and minimizes effort to obtain public documents form multiple sources coping at the same time with privacy, integrity and confidentiality issues. It simplifies the use of on top, core technologies in order to be used by all users in both the electronic and mobile environments.

Acknowledgments

The authors would like to thank the E.C. for its support in funding the SWEB [33] project and all the project partners.

References

1. "*Research in the Digital Government Realm*", special issue on the magazine of the IEEE Computer Society, 2005
2. U.S. Official e-Government portal www.firstgov.gov
3. *Local Authority Websites National Project (LAWs)*, http://www.laws-project.org.uk/
4. MMBase Relational Object Mapping Tool, http://www.mmbase.org/
5. Mioga 2 Groupware Solution, http://www.mioga2.org/en/index.html
6. *Electronic and Secure Municipal Administration for European Citizens (eMayor)*, IST-2004-507217, 6th Framework Programme, 2004, Available at: www.emayor.org
7. C. Adams, S. Lloyd, "*Understanding Public Key Infrastructure – Concepts, Standards, and Deployment Considerations*", 1st Edition, MacMillan Technical Publishing, 1999.
8. G. M. Naedele, "*Standards for XML and Web Services Security*", IEEE Computer, pp.96-98, 2003
9. A. Kaliontzoglou, P. Sklavos, T. Karantjias, D. Polemi "*A secure e-Government platform architecture for small to medium sized public organizations*", Electronic Commerce Research & Applications, Elsevier, vol. 4, No. 2, pp. 174-186, 2005
10. European Commission, "*European Interoperability Framework For Pan-European eGovernment Services*", Office for Official Publications of the European Communities, 2004, http://ec.europa.eu/idabc/servlets/ Doc?id=19528
11. OASIS Web Services Business Process Execution Language Version 2.0, WSBPEL 2.0, www.oasis-open.org/committees/wsbpel/
12. M. Fasbinder, "*Transforming Business Models to BPEL with WebSphere Business Modeler 6.02*", Technical Article, 2007, http://www.ibm.com/developerworks/ websphere/library/techarticles/0706_fasbinder/0706_fasbinder.html

13. High-powered BPEL Designer and Testing Tools for SOA Professionals, Technical Article, http://www.active-endpoints.com/active-bpel-designer.htm?eclipse
14. J. Pasley, "*How BPEL and SOA Are Changing Web Services Development*", IEEE Internet Computing, pp. 60-67, 2005
15. B. Hartman, D. J. Flinn, K. Beznosov, S. Kawamoto, "*Mastering Web Services Security*", Wiley Publishing, 2003
16. T. Dierks, C. Allen, "*The TLS Protocol Version 1.0*", RFC 2246, 1999, http://www.ietf.org/rfc/rfc2246.txt
17. WS-Security Core Specification 1.1, OASIS Standard, http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf
18. XML Signature Recommendation, XML-DSig, http://www.w3.org/TR/xmldsig-core/
19. RSA Laboratories, "*PKCS 7 v.1.5: Cryptographic Message Syntax Standard*", Technical Note, 1993, ftp://ftp.rsasecurity.com/pub/pkcs/ doc/pkcs-7.doc
20. XML Encryption, http://www.w3.org/Encryption/2001/
21. WS-Trust Specification 1.3, OASIS Standard, 2007, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.doc
22. S. Papastergiou, A. Karantjias, D. Polemi, "*A Federated Privacy-Enhancing Identity Management System (FPE-IMS)*", in The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2007.
23. M. Papazoglou, W. Heuvel, "*Service Oriented Architectures: Approaches, Technologies and Research Issues*", The VLDB Journal, Vol. 16, Issue 3, pp. 389-415, 2007
24. K. Donald, E. Varvaet, R. Stoyanchev, "*Spring Web Flow*", Reference Documentation, 2007, http://static.springframework.org/spring-webflow/docs/1.0.x/reference/index.html
25. D. Geary, C. Horstmann, "*Core JavaServer Faces*", Prentice Hall PTR, 2nd edition, 2007
26. JBoss Community, "*RichFaces framework with a huge library of rich components and skinnability support*", RichFaces Developer Guide, Red Hat, 2007, http://labs.jboss.com/file-access/default/members/jbossrichfaces/freezone/docs/devguide/en/html_single/index.html
27. R. Hightower, "*Facelets fits JSF like a glove*", Technical Article, IBM, 2006, http://www.ibm.com/developerworks/java/library/j-facelets/
28. International Organization for Standardization, "*Information Technology – Open Distributed Processing – Reference Model: Architecture*", ISO/IEC 10746-3:1996, 1996
29. W3C, "*Document Object Model (DOM)*", Technical Reports, http://www.w3.org/DOM/DOMTR
30. OASIS, "*eXtensible Access Control Markup Language (XACML) Version 1.1.*" Committee Specification, 2003, http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf
31. ETSI TS 101 903 v.1.2.2, "*XML Advanced Electronic Signatures (XAdES)*", Technical Specification, 2004, http://uri.etsi.org/01903/v1.2.2/ts_101903v010202p.pdf
32. J. Koenig, "*JBoss jBPM*", White Paper, JBoss Community, 2004, http://www.jboss.com/pdf/jbpm_whitepaper.pdf
33. *Secure, Interoperable, cross border m-services contributing towards a trustful European cooperation with the non-EU member Western Balkan countries (SWEB)*, IST-044979, 7th Framework Programme, 2006

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-295

Clustering for user modeling in recommender e-commerce application: A RUP-based intelligent software life-cycle

Anastasios Savvopoulos, Maria Virvou, Dionysios N. Sotiropoulos, George A. Tsihrintzis
University of Piraeus, Piraeus Greece

Abstract. In this paper we will present an RUP based software life cycle on how to incorporate a clustering algorithm on a prototype system. The process has four major steps. Firstly, designing and building in an adaptive application. Secondly, evaluating the system and through this process obtaining data for the clustering algorithms. Thirdly, compare the clustering algorithms with the above data as input. Fourthly, incorporate the best clustering algorithm into the system and building stereotypes. In our case we used an adaptive e-shop application as a test bed in order to apply these methods. The adaptive e-shop application that provides personalised recommendations to users.

Keywords. Clustering, user modeling, rational unified process, software life-cycle, recommendation, adaptive systems.

1. Introduction and Related Work

There are many applications incorporating recommendation techniques in order to help users in a personalized way. A very interesting technique based on a rating system has been used by Li and Kim [4]. The system acquires rates and then calculates fuzzy inferences and extracts similarities between users. Their method proved very successful according to the evaluation presented. Another approach that goes beyond attributes exploitation like the work mentioned previously is done by Alspector et.al. [9] This approach uses three adaptive techniques: feature-based, clique-based and linear model. These three techniques were combined in order to create a CART network that resulted in movie ratings. Their study showed that an effective movie-recommendation system should combine all these approaches in order to maximize performance. A lot of research effort has also been put in incorporating machine learning techniques into systems. This aim has been pursued by Castro et.al. [7] They constructed a fuzzy machine learning technique in order to obtain evident knowledge of an existing set of training examples. This technique can be used to help expert systems obtain knowledge about users and thus provide better recommendations. Their research showed that knowledge acquisition process as a process for extending, updating and improving an incomplete knowledge base, in which machine learning is useful.

Another common technique for achieving adaptivity is clustering. In this area there is work done by Kim and Ahn [6]. They constructed a GA K-means algorithm in order to create a recommender system in an online shopping market. They applied this

algorithm to a real-world case for market segmentation in electronic commerce. Another piece of research using clustering techniques concerning software segmentation, recovery and restructuring was done by Lung et.al [5]. They used a bottom–up method and clustering to group similar components together to form clusters or subsystems. Those clusters or subsystems are partitions which constitute a system. According to Lung applications of clustering analysis can be found in many disciplines but all comprise three common key steps: (1) Obtain the data set, (2) Compute the resemblance coefficients for the data set and (3) Execute the clustering method.

Despite the fact that all of the above applications of adaptive methods are very innovative and gave users results concerning adaptive help to users, none of them addressed software life-cycle issues. In adaptive applications like the ones mentioned above it's very difficult to apply the traditional software life cycle techniques. A very useful tool in software life-cycle is the Rational Unified Process (RUP). RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases: the inception, the elaboration, the construction, and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not. Moreover, one important advantage of RUP is the highly iterative nature of the development process. For the above reasons, RUP can be selected as the basis for presenting adaptive systems too.

A very novel work in incorporating the RUP technique in the life cycle of adaptive systems has been done by Kabbassi and Virvou. Their work [1] [2] [3] presents a knowledge-based software life-cycle framework for the incorporation of multicriteria analysis in intelligent user interfaces. In all their work they use RUP as the tool of designing and developing their system. The usage of RUP proved to be very efficient on designing an adaptive system and enhanced the software life –cycle process of these systems.

In this paper we will present an RUP based software life cycle on how to incorporate a clustering algorithm on a prototype system. The process has four major steps. First, designing and building the prototype system that does not include any clustering techniques. Second, evaluating the system and through this process obtaining data for the clustering algorithms. Third, comparing several clustering algorithms with the above data as input and choosing the most efficient algorithm. Fourth, incorporating the clustering algorithm into the system and building stereotypes based on this algorithm. In our case we used an adaptive e-shop application as a test bed in order to apply these methods called Vision.Com [16].

2. Creating the Software Life Cycle

The Vision.Com life – cycle framework is based on RUP and is presented in the table below. RUP gives a framework of a software life-cycle that is based on iterations. However, RUP neither specifies what sort of requirements analysis has to take place for adaptive systems nor what kind of prototype has to be produced during each phase or procedural step. In this table, we have maintained the phases and procedural steps of RUP. Based on this, we have specified what prototype has to be constructed and what

kind of experiment has to be conducted. Therefore, this table represents our solution to this problem.

Procedural Steps/Phases	Inception	Elaboration				Construction	Transition
Requirements Capture	Requirements of a prototype adaptive recommender system without clustering.					The most efficient clustering algorithm.	
Analysis & Design	Analysis and Design of the prototype adaptive recommender system without clustering.	Computing the resemblance coefficients for the data set and developing the clustering algorithm.				Designing double stereotypes resulted from the selected clustering algorithm.(of users and products)	
Implementation	Building the prototype adaptive recommender system without clustering.	Execute the clustering method for the prototype.				Building the user modeling component based on the stereotypes and incorporating them into the system.	Dynamically improving system performance while used by real users.
Testing	Evaluating the system and obtaining the data set for the clustering techniques.	Evaluating the Results of the clustering algorithm used in the prototype.				Comparing the results provided with those of the prototype system.	
Iterations	Iter #1	Iter #2	Iter #3	…	Iter #n	Iter #n+1	Iter #n+2

Table 1 Presenting the RUP steps.

3. Inception

3.1. Defining Requirements for the prototype system and Analysis and Design of the prototype adaptive recommender system

In our days marketplace is filled with e-commerce systems that try to sell customers products they really need. But a large number of these systems do not address customers' specific needs and interest. Our aim was to create a novel system that would take in mind customers' needs and provide them with product recommendations that really suit their interests.

In order to design our system according to the initial requirements we used UML technology in every step of the RUP to help us define classes, objects and relationships between them. In this section we present a use case diagram of the prototype system operations. In the diagram below we see the recommender module as an actor. From top to bottom the recommender can increase the interest degree of specific category if a user visits this category, can increase all movie features if a user decides to visit a specific movie or place it to his/her cart or buy it and can extract recommendations from the user model when the user visits the recommendation page. When a new user registers the recommender module can extract information from his/her questionnaire

during his/her registration and place this user to the assorted stereotype. Lastly the recommender controls the animated agent of the system and tries to provide the user with useful tips concerning movies and their features.

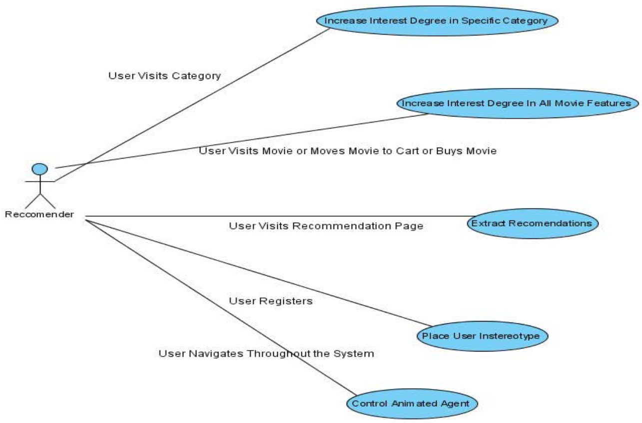


Figure 1 Use Case Diagram of the Recommender

3.2. Building the prototype adaptive recommender system

Vision.Com (fig. 2) is an adaptive e-commerce video store that learns from customers' preferences. Its aim is to provide help to customers choosing the best movie for them. Vision.Com operates on the top of the .net framework technology and uses client server techniques. The rule-based user model built on the system was formed by rules on what a customer usually considers when choosing a movie. More information on Vision.Com prototype can be found in [16].

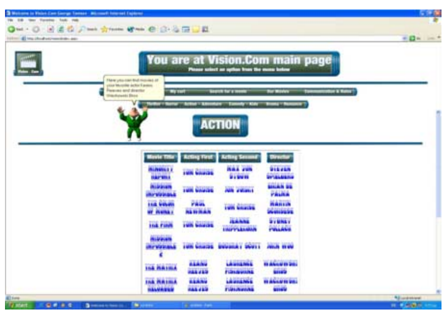


Figure 2 Vision.Com User Interface

3.3. Evaluating the system and obtaining the data set for the clustering techniques

The experiment was conducted taking into account [10] in order to be more efficient and for the results to be drawn without any compulsion. First of all, we used a significant number of students to use and interact with the system. They were 80 persons that used the system. These students were divided into three groups that conducted the experiment into three different time zones. More specifically the experiment was conducted in three different days and different hours. The experiment

for each user was consisted of three sections. First the participant is asked to visit a significant amount of movies that he has different degrees of interest in them. Secondly he must choose enough of them to put into his cart. In the second section, he must go to his cart and choose to delete or not movies that he had chosen from his cart for any reason (including the price of the movie). At the last section he must choose to buy these movies that he concluded to have in his cart. In the end every participant was asked questions about the systems visual appearance, usage, effectiveness and adaptivity in his tastes.

4. Elaboration

4.1. Computing the resemblance coefficients for the data set and developing the clustering algorithm

Despite the fact that evaluation results of the original system were very promising, they also revealed problems concerning the clustering of similar users and making right recommendation based on small differences. More specifically stereotypes didn't prove very effective to categorize users as they were based on strict categories of users and didn't change dynamically as the customer used the system. Furthermore, the rules that Vision.Com used to make assumptions and provide the customers with product recommendations proved many times to be inadequate. Especially, in cases of small differences in customers tastes.

To address these requirements we decided to integrate a clustering algorithm to our system in order to strengthen the recommendation module. We conducted a comparison between four clustering algorithms in order to find out which one provided us with the best results. The input data for the algorithms were taken from experimental results of the evaluation. In this work, we tested and compared four clustering techniques, namely a) agglomerative hierarchical clustering [11], b) fuzzy c-means clustering [12] and c) spectral clustering [13] against AIS-based clustering [14], [15].

4.2. Execute the clustering method for the prototype and Evaluating the Results of the clustering algorithm used in the prototype

The data collected from the experiment consisted of three parts. Every part is similar to the others. The first one contains statistical data of the visits that every user made to specific movies. The second part contains data of the moves the users moved into their cart. The last part contains statistical data concerning the preferences on the movies bought by every user. Every record in every part is a vector of the same 80 features that were extracted of the movies' characteristics and represents the references of one user. The features include all the categories, all the sets of prices, all the actors and all the directors that are known to the system.

These 80 feature vectors were used to represent both individual users' interests and movie descriptions. In particular, in the case individual users' interests every constituent in the 80 – dimensional vectors takes as a value the mean percentage of the user's interest in this particular feature according to Equation 1. In the case of the representation of movie features the constituents of 80-dimentional vectors that apply to a particular movie take the value of 1 otherwise they take the value 0. For example, a movie directed by Copolla takes the value of 1 in this feature and takes the value of

zero in all other directors. We used these final 80 - dimensional vectors as an input to four different clustering algorithms a) agglomerative hierarchical clustering, b) fuzzy c-means clustering and c) spectral clustering against AIS-based clustering called AIN. This was done in order compare the results of the first three with the last and to prove that AIN excels against the others.

In Fig. 3, we show and compare dendrograms corresponding to the feature vectors produced by hierarchical (top left), spectral (top right), and AIN-based (centre left) clustering, respectively. We observed that spectral clustering does not provide a clearer revelation of the intrinsic similarities in the dataset over hierarchical clustering. On the other hand, the leaves of AIN-based dendrogram are significantly fewer than the leaves of the other dendrograms in Fig. 3, which stems from the fact that the former corresponds to clustering only 22 representative points in the 80-dimensional feature space, while the latter two correspond to clustering the complete set of 80 data points. Thus, the AIN-based dendrogram demonstrates the intrinsic data point clusters significantly more clearly and compactly than the corresponding hierarchical and spectral dendrograms.

Also, in Fig. 3, we show the partitioning of the complete dataset into six clusters by the spectral (centre right), fuzzy c-means (bottom left), and AIN-based (bottom right) clustering algorithms. We observe that spectral clustering does not result in cluster homogeneity, while fuzzy c-means clustering results in higher cluster homogeneity, but in only four clusters rather than six required. Specifically, we observed that fuzzy c-means clustering assigned the same degree of cluster

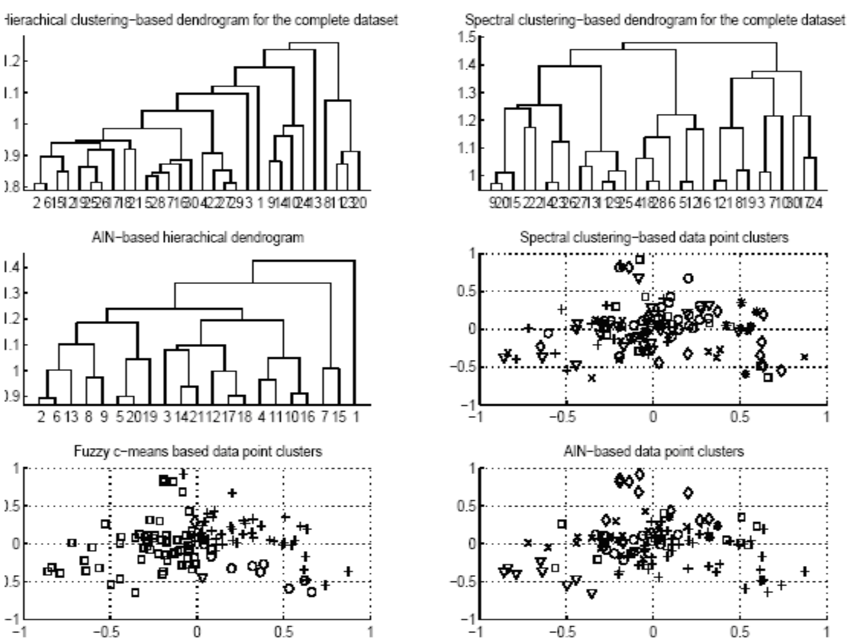


Figure 3 Result diagrams of the clustering methods

membership to all the data points, which implies that certain intrinsic data dissimilarities were not captured by the fuzzy c-means clustering algorithm. On the contrary, AIN-based clustering returned significantly higher cluster homogeneity. Moreover, the degree of intra-cluster consistency is significantly higher in the AIN-based rather than the hierarchical and spectral clusters, which is of course a direct consequence of a data redundancy reduction achieved by the AIN. Fig. 3 leads to the conclusion that Vision.Com customers exhibit certain patterns of behaviour when shopping and tend to group themselves into six clusters. The 22 antibodies that arose via the AIN-based clustering algorithm correspond to customer behaviour representatives and, thus, can be seen as important customer profiles, which eventually correspond to stereotypes in user models.

5. Construction

5.1. The most efficient clustering algorithm

After deciding that AIN based clustering suited better for our system we redesigned and built out system integrating AIN clustering into it. The new model works as follows: At the initial state the system includes all user model data of all users. This data is peddled to the AIN algorithm that produces clusters of similar users. Every cluster has a center or more that is the representative of this cluster. The centers are called antibodies. In order to connect a user with an antibody the system calculates the minimum Euclidian distance between the antibodies and the user. Then with the help of the antibody finds the minimum distances between movies vectors and antibody vector. The movies with the minimum distance are rated as very similar and presented to this specific user as the recommended ones with the help of adaptive hypermedia.

5.2. Designing double stereotypes resulted from the selected clustering algorithm (of users and products)

The AIN based Vision.Com produced very good results concerning product recommendations and general recommendations throughout the navigation of the system for every individual customers. The more a customer used Vision.Com the better were the recommendations. But our system lacked at producing recommendations to new customers and customers that had little knowledge of, in general. In order to dissolve this problem we decided to create dynamic double stereotypes. Stereotypes for customers and stereotypes of movies based on the AIN clustering that varied in complexity and accuracy. The state diagram shows the states a new user can pass until the system classifies him as old user. This is the initialization process of Vision.Com in order to achieve the point of adequate information about this specific user. Every level of stereotype that the user passes to is more complex and involves more movie features. The last level includes all movie features and the user is categorized as old.

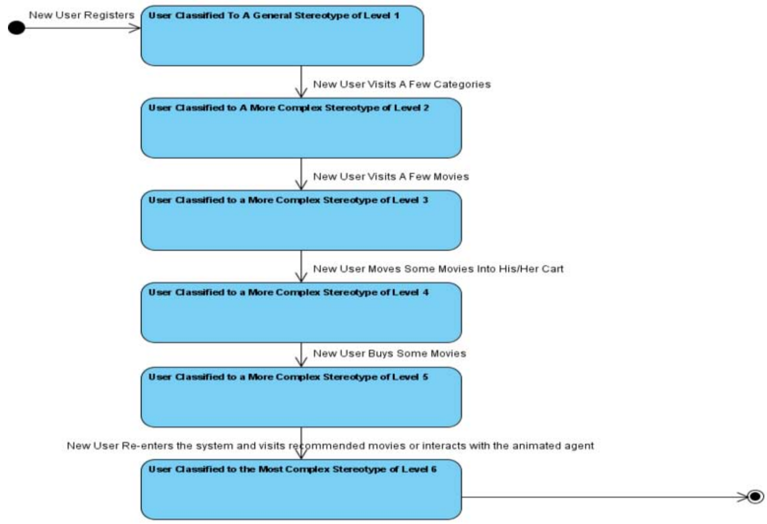


Figure 4 State diagram of the initialization process

5.3. Building the user modeling component based on the stereotypes and incorporating them into the system

The process described above led to a double system of stereotypes for both users and movies. In this way, selling movies in the e-shop could be performed more efficiently. The efficiency was provided by the fact that the system not only new information about the customers but the movies as well. Specifically, for every customer there were individual suggestions that were produced very quickly.

Moreover, we achieved a quick incremental initialization of the user model meaning that the video store could respond to customer needs in minimum amount of time. The double classification (users' interests – movies) was performed in a hierarchical way that resulted in several levels of user stereotypes which corresponded to the clusters that were described in the previous section. Thus we constructed a hierarchy of stereotypes, as illustrated in Figure 5. At first, there was a coarse classification of two stereotypes which was next refined several times to produce a final classification of six stereotypes. These stereotypes are then used dynamically by the e-commerce application to infer users' interests in movies based on a small set of observed users' actions. In fact, for a new user, modelling is performed based on the first classification of users. Then, incrementally and while the user interacts with the system, inferences about his/her preferences are drawn from more refined user stereotypes of subsequent levels.

The actual construction of stereotypes mainly involves defining the triggering conditions (the conditions that enable a specific stereotype), and the inferences (what can be assumed for users belonging in the triggered stereotype). Whether a customer or a movie belongs or not to a stereotype is measured by the Euclidean distance. If a customer's distance is very far from the leaves then we proceed to the next level above (figure 5). If in this level the distance is also far we move up to the next. This process is continued until we reach the top level or the certain amount of distance that is considered close enough. When this distance is reached the system follows the same process for movies.

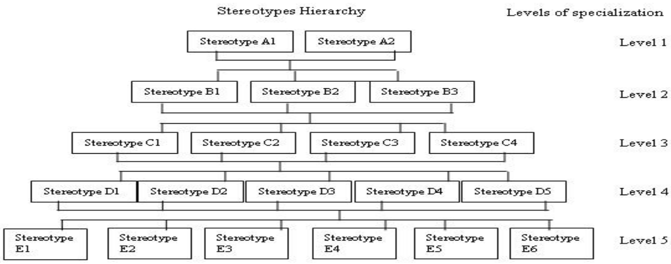


Figure 5 Hierarchy of stereotypes

6. Transition-Dynamically improving system performance while used by real users.

When a new user becomes a member to the system the e-commerce application creates a profile, sets all interest values into zero (the system assumes that at start the user has no interest for any movie) and starts to monitor his/her actions. After few interactions with the system (visiting few movies), vision.com classifies the new user in a stereotype of the first level of specialization. The first level generally checks interest concerning the four movie categories. For example, if the new user shows a tendency towards thriller movies them s/he classified in the second stereotype, as the main difference between the two stereotypes can be seen in this movie category. If the user belongs to the first stereotype the system chooses to propose movies from any of the three categories except thrillers in a presentation percentage similar to the interest in every category. The movies that are most close to this movie stereotypes are those recommended by the system up to this point of inter-action. As the user continues with moving movies into his/her cart and buying some of them the system moves to the next level of classification that extends stereotypical information to the price features. This means that in this level stereotypes differ greatly in the price ranges along with movie categories. For example, if the new user selects movies with medium prices the system chooses to classify him/her to the stereotype that has the greatest interest in this price range concerning always the interest in movie categories.

Level four and five of specialization extend the features of interests into interests in leading actors and directors accordingly. In this way, as users show with their actions which actor or director prefer the system easily classifies them into the according stereo-types and selects the right movie stereotypes in order to make recommendations. If their differences in interest in the actor and directors are low the system chooses to group users in same stereotypes thus emulating the grouping process into the previous level of specialization. On the other hand if these differences are high the users are grouped into different stereotypes of these levels. The initialization process is conducted until the user reaches level six of specialization. This level represents the leaves in the hierarchical tree of stereotypes and extends the differences in all features of interests. The specialization in these level is very high and even the smallest difference in user's interest can classify him/her to a different stereotype of this specific level.

7. Conclusions

In this paper we presented an RUP based software life cycle on how to incorporate a clustering algorithm in an adaptive and personalised application. We used an e-shop application as a test bed in order to apply these methods. Through the developing and evaluating procedures we showed that RUP can be selected as the basis for a life-cycle procedure that leads to the incorporation of clustering methods in user modelling.

References

[1] A Knowledge-Based Software Life-Cycle Framework For The Incorporation Of Multicriteria Analysis In Intelligent User Interfaces Katerina Kabassi And Maria Virvou, Member, Ieee
[2] Experimental Studies Within Software Engineering Process For Intelligent Assistance In A Gui. Mariva Virvou Katerina Kabassi
[3] An Object - Oriented Approach In Knowledge Based Software Engineering Of An Intelligent Guị Maria Virvou, Katerina Kabassi
[4] Constructing User Profiles For Collaborative Recommender System Qing Li And Byeong Man Kim
[5] Applications Of Clustering Techniques To Software Partitioning, Recovery And Restructuring Chung-Horng Lung, Marzia Zaman, Amit Nandi
[6] A Recommender System Using Ga K-Means Clustering In An Online Shopping Market Kyoung-Jae Kim, Hyunchul Ahn
[7] Use Of A Fuzzy Machine Learning Technique In The Knowledge Acquisition Process J.L. Castroa, J.J. Castro-Schezb, J.M. Zuritaa
[9] Feature-Based And Clique-Based User Models For Movie Selection: A Comparative Study Joshua Alspector, Aleksander Kołcz And N. Karunanithi
[10] DN Chin. Empirical Evaluation of User Models and User-Adapted Systems. User Modeling and User-Adapted Interaction, 2001 - Springer
[11] Efficient algorithms for agglomerative hierarchical clustering methods WHE Day, H Edelsbrunner - Journal of Classification, 1984 - Springer
[12] FCM: The fuzzy c-means clustering algorithm. Bezdek, J C | Ehrlich, R | Full, W COMP. GEOSCI. Vol. 10, no. 2-3, pp. 191-203. 1984
[13] On spectral clustering: Analysis and an algorithm A Ng, M Jordan, Y Weiss - Advances in Neural Information Processing Systems 14: , 2002.
[14] Cayzer, S., Aickelin, U.: A Recommender System based on the Immune Network. Proceedings of the 2002 Congress on Evolutionary Computation (2002)
[15] Morrison, T., Aickelin, U.: An Artificial Immune System as a Recommender for Web Sites. Proceedings of the 1st Conference on ARtificial Immune Systems (ICARIS-2002) Canterbury UK (2002) 161–169
[16] User modelling for adaptive e-shopping (Maria Virvou & Savvopoulos Anastasios), 11th International Conference on Human Computer Interaction (HCII 2005), 22-27 July, Las Vegas, Nevada, USA.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-305

Knowledge Engineering for Affective Bi-Modal Interaction in Mobile Devices

Efthymios ALEPIS[1], Maria VIRVOU[1], Katerina KABASSI[2]

[1]*Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou St., 18534 Piraeus, Greece*

[2]*Department of Ecology and the Environment, TEI of the Ionian Islands, 2 Kalvou Sq, 29100, Zakynthos, Greece*

E-mail: talepis@unipi.gr; mvirvou@unipi.gr; kkabassi@teiion.gr

Abstract: This paper focuses on knowledge engineering for the development of a system that provides affective interaction in mobile devices. The system bases its inferences about users' emotions on user input evidence from the keyboard and the microphone of the mobile device. For this purpose different experimental studies have been conducted with the participation of mobile users and human experts. The experiments' aim was twofold. They aimed at revealing the criteria that are taken into account in each mode for emotion recognition as well as their weight of importance. The results of the studies are further used for the application of a multi-criteria decision making model.

1. Introduction

In the last decade, there is a growing interest in mobile technology and mobile networks. As a result, a great number of services are offered to the users of mobile phones. One important field where mobile technology can make significant contributions is education. A recent review has shown that all well known technologies from the areas of ITSs (Intelligent Tutoring Systems) have already been re-implemented for the Web.

In the fast pace of modern life, students and instructors would appreciate using constructively some spare time. They may have to work on lessons at any place, even when away from offices, classrooms and labs where computers are usually located. At the current state, there are not many mature mobile tutoring systems since the technology of mobile computing is quite recent and has not yet been used to the extent that it could. However, there have been quite a lot of primary attempts to incorporate mobile features to this kind of educational technology and the results so far confirm the great potential of this incorporation. Moreover, in many cases it would be extremely useful to have such facilities in handheld devices, such as mobile phones rather than desktop or portable computers so that additional assets may be gained. Such assets include device independence as well as more independence with respect to time and place in comparison with web-based education using standard PCs.

However, different problems may occur during people's interaction with mobile devices. This is especially the case of novice users who find such an interaction frustrating and difficult. A remedy to such problem may be given by providing adaptive interaction based on the user's emotional state. The recognition of emotions can lead to affective user interfaces that take into account the users' feelings and can adapt their behavior according to these feelings. Regardless of the various emotional paradigms, neurologists/psychologists have made progress in demonstrating that emotion is at least as and perhaps even more important than reason in the process of decision making and action deciding [1]. Moreover, the way people feel may play an important role in their cognitive processes as well [2]. Recently, significant research effort has been put in the recognition of emotions of users while they interact with software applications. Picard points out that one of the major challenges in affective computing is to try to improve the accuracy of recognizing people's emotions [3].

Improving the accuracy of emotion recognition may imply the combination of many modalities in user interfaces. Indeed, human emotions are usually expressed in several ways. Human faces, people's voices, or people's actions may all show emotions. Ideally, evidence from many modes of interaction should be combined by a computer system so that it can generate as valid hypotheses as possible about users' emotions. It is hoped that the multimodal approach may provide not only better performance, but also more robustness [4].

In previous work, the authors of this paper have implemented and evaluated with quite satisfactory results emotion recognition systems, incorporated in educational applications ([5], [6]). As a next step we have extended our affective educational system by providing mobile interaction between the users and the system. There are many virtues of web-based educational software, which have been recognized by educators and educational institutions. Some important assets include platform-, place-, and/or time-independence. In many situations this means that learning may take place at home or some other site, supervised remotely and asynchronously by a human instructor but away from the settings of a real class.

However, in many cases it would be extremely useful to have such facilities in handheld devices, rather than desktop or portable computers so that users could use the software on a device that they can carry anywhere they go. Handheld devices render the software usable on every occasion, even when one is standing rather than sitting. However, among handheld devices, which include palm or pocket PCs and mobile phones, the mobile phones provide the additional very important asset of computer-device independence for users.

In view of the above, in this paper we describe knowledge engineering for a novel mobile educational system that incorporates bi-modal emotion recognition. The two modes of interaction are the interaction through the mobile device's keyboard and the interaction through the users' voice. Users may use their mobile device in order to read parts of the theory about a particular lesson, as well as to take tests and sort examinations. In the case of the sort examinations about particular lessons, users may write their answers directly to their mobile device through the keyboard, or in other cases they may use the mobile device's microphone as a mode of interaction. The proposed system collects evidence from the two modes of interaction and analyses them in terms of criteria for emotion recognition. The main focus of the paper is on presenting the empirical studies that are essential for further application of a multi-criteria model, which is going to be used for making final assumptions about the recognition of one or more than one emotional states.

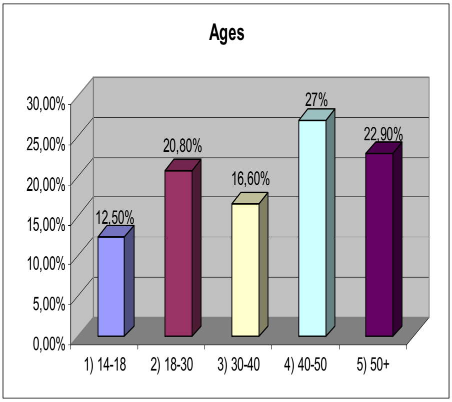
2. Affective interaction in Mobile Devices

After a thorough investigation in the related scientific literature we found that there is a shortage of educational systems that incorporate multi-modal emotion recognition. Even less are the existing affective educational systems with mobile facilities. In [7] a mobile context-aware intelligent affective guide is described, that guides visitors touring an outdoor attraction. The authors of this system aim mainly in using a mobile guide that produces emotions to the users. On the contrary our proposed system aims at recognizing the users' emotions through their interaction with a mobile device. As a second related approach we found that Yoon et al. [8] propose a speech emotion recognition agent for mobile communication service. This system tries to recognize five emotional states, namely neutral emotional state, happiness, sadness, anger, and annoyance from the speech captured by a cellular phone in real time and then it calculates the degree of affection such as love, truthfulness, weariness, trick, and friendship. In this approach only data from the mobile device's microphone are taken into consideration, while in this paper we investigate a mobile bi-modal emotion recognition approach. Moreover, our proposed system is incorporated in an educational application and data pass through a linguistic and also a paralinguistic stage of analysis. This derives from the fact that in an educational application we should take into consideration how users say or write something (such as low or high voice, slow or quick write speed), as well as what users say or write (such as correct answers or mistakes).

3. Requirement Analysis for Determining Criteria

The empirical study that we have conducted concerns the audio-lingual emotion recognition, as well as the recognition of emotions through keyboard evidence. The audio-lingual mode of interaction is based on using a mobile device's microphone as input device. The empirical study aimed at identifying common user reactions that express user feelings while they interact with mobile devices. As a next step, we associated these reactions with particular feelings.

The empirical study involved a total number of 100 male and female users of various educational backgrounds, ages and levels of familiarity with computers. Individuals' behaviour while doing something may be affected by several factors related to their personality, age, experience, etc. Figure 1.a illustrates the distribution of participants in the empirical study in terms of their age. In particular there were 12,5 % of participants under the age of 18, approximately 20% of participants between the ages of 18 and 30. A considerable percentage of our participants was over the age of 40. Figure 1.b illustrates the distribution of participants in the empirical study in terms of their computer knowledge level.



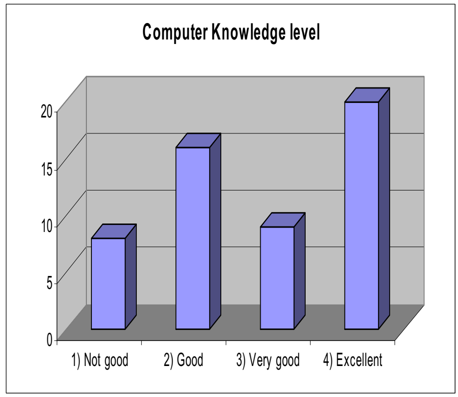


Figure 1.a Ages of participants
Figure 1.b Computer knowledge level of the participants

The participants were asked to use a mobile application which incorporated a user monitoring component. The user monitoring component that we have used can be incorporated in any application, since it works in the background recording each user's input actions. However, an application that generates a sufficient number of user emotional states is more appropriate for our purposes. Therefore, for the purposes of the empirical study, we incorporated it into an educational application. Part of the interaction included knowledge tests, while participants were asked to use oral interaction via their mobile device's microphone. Our aim was not to test the participants' knowledge skills, but to record their oral and written behaviour. Thus, the educational application incorporated the monitoring module that was running unnoticeably in the background. Moreover, users were also video taped while they interacted with the mobile application.

After completing using the educational application, participants were asked to watch the video clips concerning exclusively their personal interaction and to determine in which situations they where experiencing changes in their emotional state. Then, they associated each change in their emotional state with one of the six basic emotion states in our study and the data was recorded and time-stamped.

As the next step, the collected transcripts were given to 20 human expert-observers who were asked to perform audio emotion recognition with regard to the six emotional states, namely happiness, sadness, surprise, anger, disgust and neutral. All human expert-observers possessed a first and/or higher degree in Psychology and, to analyze the data corresponding to the audio-lingual input only, they were asked to listen to the video tapes without seeing them. They were also given what the user had said in printed form from the computer audio recorder. The human expert-observers were asked to justify the recognition of an emotion by indicating the weights of the criteria that they had used in terms of specific words and exclamations, pitch of voice and changes in the volume of speech.

Finally after processing the data from both the human experts and the monitoring component we came up with statistical results that associated user input action through the mobile keyboard and microphone with emotional states. More specifically, considering the keyboard we have the following categories of user actions: a) user types normally b) user types quickly (speed higher than the usual speed of the

particular user) c) user types slowly (speed lower than the usual speed of the particular user) d) user uses the “delete” key of his/her mobile device often e) user presses unrelated keys on the keyboard f) user does not use the keyboard. These actions are considered as criteria for the evaluation of emotion with respect to the user’s action in the keyboard.

Considering the users’ basic input actions through the mobile device’s microphone we have 7 cases: a) user speaks using strong language b) users uses exclamations c) user speaks with a high voice volume (higher than the average recorded level) d) user speaks with a low voice volume (low than the average recorded level) e) user speaks in a normal voice volume f) user speaks words from a specific list of words showing an emotion g) user does not say anything. These seven actions are considered as criteria for the evaluation of emotion with respect to what the user says.

Concerning the combination of the two modes in terms of emotion recognition we came to the conclusion that the two modes are complementary to each other to a high extent. In many cases the human experts stated that they could generate a hypothesis about the emotional state of the user with a higher degree of certainty if they had taken into account evidence from the combination of the two modes rather than one mode. Happiness has positive effects and anger and boredom have negative effects that may be measured and processed properly in order to give information used for a human-computer affective interaction. For example, when the rate of pressing the mobile device’s “deletion key” of a user increases, this may mean that the user makes more mistakes due to a negative feeling. However this hypothesis can be reinforced by evidence from speech if the user says something bad that expresses negative feelings.

4. Requirement Analysis for Determining the Weights of the Criteria

The previous empirical study revealed the criteria that are taken into account when evaluating different emotions. However, these criteria are not equally important for evaluating different emotions. For this purpose another empirical study was conducted. More specifically, the human experts who participated in the first empirical study and selected the final set of criteria were also asked to rank the 13 criteria with respect to how important they are in their reasoning process. First, the experiment focused on the 6 input actions of the keyboard and then to the seven input actions of the microphone.

Therefore, each human expert was asked to share 21 points into the 6 different criteria with respect to the keyboard input for the emotion of happiness. As soon as the scores of all human experts were collected, they were used to calculate the weights of criteria. The scores assigned to each criterion by all human experts were summed up and then divided to the sum of scores of all criteria (21 points assigned to all criteria by each human expert * 20 human experts = 420 points assigned to all criteria by all human experts). In this way the sum of all weights could be equal to 1.

As a result, the calculated weights for the criteria were the following:

- The weight for the criterion k1:

$$w_{e_1 k1} = \frac{93}{420} = 0.22$$

- The weight for the criterion k2:

$$w_{e_1 k2} = \frac{116}{420} = 0.28$$

- The weight for the criterion k3:

$$w_{e_1k3} = \frac{81}{420} = 0.19$$

- The weight for the criterion k4:

$$w_{e_1k4} = \frac{43}{420} = 0.10$$

- The weight for the criterion k5:

$$w_{e_1k5} = \frac{65}{420} = 0.15$$

- The weight for the criterion k6:

$$w_{e_1k6} = \frac{22}{420} = 0.05$$

As soon as this process was completed for the emotion of happiness, it was repeated for all the other emotions. As a result, there was a set of weights for the criteria that correspond to the keyboards' input actions for each different emotion.

Then each human expert was asked to share 28 points into the 7 different criteria with respect to the microphone input for the emotion of happiness. For example, one human expert thought the criteria m3 and m6 were equally important to each other but more important compared to all the others. Therefore, s/he assigned 7 points to each one. S/he also thought that the second most important criterion was m2 and assigned 6 points to it. Furthermore, s/he assigned 3 points to m4 and m5 as they were considered as equally important. Finally, s/he assigned the rest of their 2 points equally in the other two criteria, namely m1 and m7, which s/he considered as less important of all.

As soon as the scores of all human experts were collected, they were used to calculate the weights of criteria. The scores assigned to each criterion by all human experts were summed up and then divided to the sum of scores of all criteria (28 points assigned to all criteria by each human expert * 20 human experts = 560 points assigned to all criteria by all human experts). In this way the sum of all weights could be equal to 1.

As a result, the calculated weights for the criteria were the following:

- The weight for the criterion m1:

$$w_{e_1m1} = \frac{20}{560} = 0.04$$

- The weight for the criterion m2:

$$w_{e_1m2} = \frac{123}{560} = 0.22$$

- The weight for the criterion m3:

$$w_{e_1m3} = \frac{104}{560} = 0.19$$

- The weight for the criterion k4:

$$w_{e_1m4} = \frac{59}{560} = 0.11$$

- The weight for the criterion m5:

$$w_{e_1m5} = \frac{82}{560} = 0.15$$

- The weight for the criterion m6:

$$w_{e_1m6} = \frac{138}{560} = 0.25$$

- The weight for the criterion m7:

$$w_{e_1m7} = \frac{34}{560} = 0.06$$

As soon as this process was completed for the emotion of happiness, it was repeated for all the other emotions. As a result, there was a set of weights for the criteria that correspond to the microphones' input actions for each different emotion. For the emotion of happiness we may notice the leading importance of criterion m6, considering the interaction through the mobile device's microphone. In this case a user has spoken words from a specific list of words that is considered to reveal an emotional state. Such a case could be the use of the word "great" by a user, which would express her/his satisfaction in certain phases of her/his interaction with the mobile system. The use of this specific criterion results from the paralinguistic data analysis (what the user actually says) and is of great importance in the recognition of all six basic emotional states of our study. Other criteria, like the first two in the interaction through the keyboard, reveal easiness in writing and may indicate positive effects in a users' emotional state.

5. Overview of the interaction through the Mobile Device

The main architecture of the mobile bi-modal emotion recognition system is illustrated in figure 2. Participants were asked to use their mobile device and interact with a pre-installed educational application. Their interaction could be accomplished either orally (through the mobile device's microphone) or by using the mobile devices keyboard and of course by combining these two modes of interaction. All data are captured during the interaction of the users with the mobile device through the two modes of interaction and then transmitted wirelessly to the main server. All the input actions described in section 3 are used as trigger conditions for emotion recognition by the emotion detection server. Finally all input actions as well as the possible recognized emotional states are stored in the system's database.

The discrimination between the participants is done by the application that uses the main server's data base and for each different user a personal profile is created and

stored in the data base. In order to accomplish that, user name and password is always required to gain access to the mobile educational application.

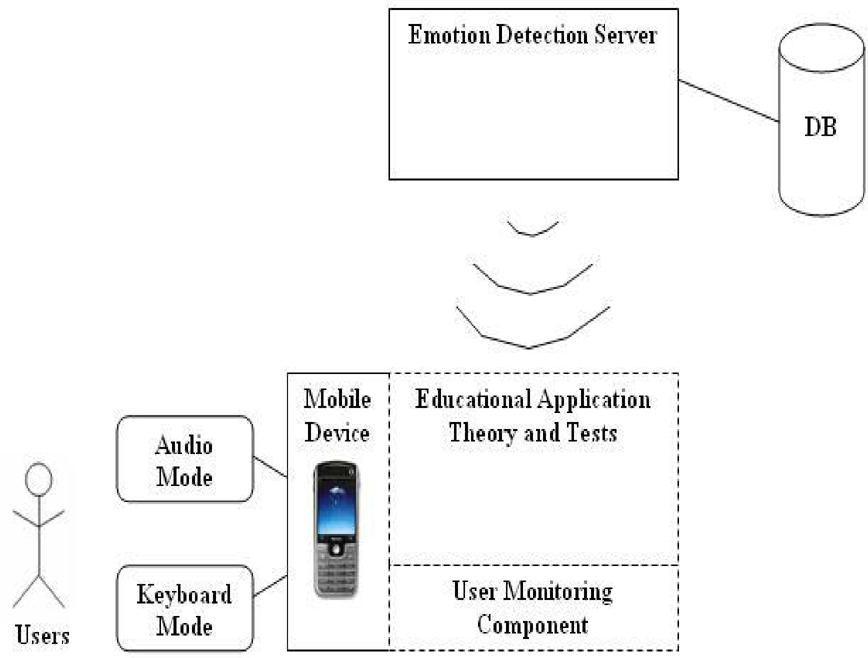


Figure 2. Architecture of the mobile emotion detection system.

A snapshot of a mobile emulator, operated by a participant is illustrated in figure 3. Users may answer questions and take tests using the mobile system. They may write their answers through the mobile device's keyboard, or alternatively give their answers orally, using their mobile device's microphone. In both cases, the data from the two possible modes of interaction are stored in the main system's database (emotion detection server), in order to be processed for emotion recognition purposes. When participants answer questions, the system tries to perform error diagnosis in cases where the participants' answers have been incorrect. Error diagnosis aims at giving an explanation about a participant's mistake taking into account the history record of the participant and the particular circumstances where the error has occurred. Giving a correct explanation of a mistake can be a difficult task for the system. One problem that complicates this task further is ambiguity, since there may be different explanations of observed incorrect users' actions. For example, in a fill-in the blank space question a participant may give an incorrect answer simply because s/he has mistyped the answer. In this case this may indicate a consequence of a negative emotional state. However, this may well appear as a lack of knowledge in the domain being taught.

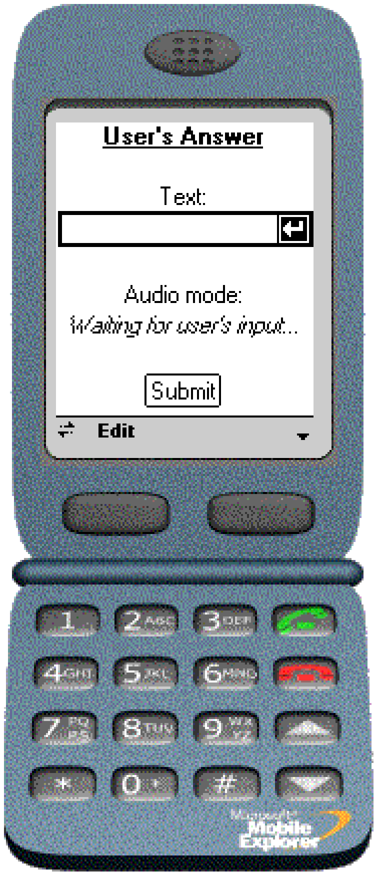


Figure 3. A user is answering a question of a test either using the keyboard or orally through the mobile device's microphone.

6. Conclusions

In this paper we have described the empirical studies conducted during knowledge engineering for an affective bi-modal system for mobile devices. More specifically, two different experiments are described. The first experimental study participated real users as well as human experts and aimed at revealing the criteria that are taken into account for emotion recognition in mobile devices.

However, the criteria, which the first experiment revealed, were not equally important in emotion recognition. In fact they have different weights of importance in the recognition of different emotions. Therefore, a second experimental study with the participation of human experts was conducted and its results revealed the weights of importance of the criteria mentioned above for six different emotions.

The results of these experiments revealed the criteria used for emotion recognition in mobile devices were similar to the criteria that were identified by other experimental studies [5] for emotion recognition in personal computers. As a result, these findings

may also be used by other systems that perform emotion recognition in other domains. However, what may differ in the application of these criteria in other systems is their weight of importance in emotion recognition. In such a case, the setting of the second experiment should be repeated for the new domain.

7. References

[1] Leon E, Clarke G, Gallaghan V, Sepulveda F, "A user-independent real-time emotion recognition system for software agents in domestic environments", Engineering applications of artificial intelligence, 20 (3): 337-345 APR 2007

[2] Goleman, D.: Emotional Intelligence, Bantam Books, New York (1995)

[3] Picard, R.W.: Affective Computing: Challenges, Int. Journal of Human-Computer Studies, Vol. 59, Issues 1-2, (2003) 55-64

[4] Pantic, M., Rothkrantz, L.J.M.: Toward an affect-sensitive multimodal human-computer interaction. Vol. 91, Proceedings of the IEEE (2003) 1370-1390

[5] Alepis, E., Virvou, M. and Kabassi, K.: "Knowledge Engineering for Affective Bi-modal Human-Computer Interaction, Sigmap (2007)

[6] Alepis, E. and Virvou, M.: Emotional Intelligence: Constructing user stereotypes for affective bi-modal interaction, KES (2006)

[7] Lim, M.Y., Aylett, R., Feel the difference: A guide with attitude!, Lecture Notes in Computer Science, Volume 4722 LNCS, 2007, Pages 317-330

[8] Yoon, W.-J., Cho, Y.-H., Park, K.-S., A study of speech emotion recognition and its application to mobile services, Lecture Notes in Computer Science, Volume 4611 LNCS, 2007, Pages 758-766

Knowledge Engineering for Process Management and Project Management

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-317

Analogy Based Cost Estimation Configuration with Rules[1]

Stamatia BIBI[2], Ioannis STAMELOS
Department of Informatics, Aristotle University of Thessaloniki, Greece

Abstract. Analogy-based estimation is a widely adopted method in software cost estimation that identifies analogous projects to the one under estimation and uses their data to derive an estimate, i.e. it is a Case Based Reasoning approach. The similarity measures between pairs of projects are critical for identifying the most appropriate historical data from which the estimation will be generated. Usually the similarity measures are selected empirically, using jackknife-like procedures. Typically, the measures that identify the most similar projects in most of the cases are considered the most appropriate ones and are applied in every new estimation procedure. However there are situations that the default similarity measures may not be the most appropriate ones. In this study we determine the situations in which the default parameters are not the best and we propose the similarity measures for these cases. In particular we provide rules that point out which projects are not accurately estimated with the default parameters.

Keywords. Analogy, Software Cost estimation, Case based reasoning, Rules,

1. Introduction

Software cost estimation is the process of predicting the amount of effort required to develop a software system. Usually the estimation is performed before the initialization of a project and is utilized throughout all software lifecycle, determining the feasibility of a project, the project plan, the allocation of resources and finally the project progress. Accurate and consistent estimates are fundamental to several success-critical project factors, justifying the existence of a variety of estimation models [4], [5],[6], [14].

It is common practice in almost all automated estimation models to use past historical data of already completed projects to predict future ones. One of the most common methods that utilizes past project data is Estimation based on Analogy (EBA) [12]. EBA identifies one or more historical projects that are similar to the project being developed and, based on the data of these projects, derives an estimate. Frequent application of EBA [9] in software cost estimation has indicated certain advantages of the method. EBA can be applied in the early stages of software development when few

[1] This work has been funded by the Greek Secretary of Research and Technology, action 4.3.1, DIERGASIA project.

[2] Corresponding author

data are available and it produces results, easily interpreted by software managers that opposed to other formal models remain unaffected by outliers.

However, there are also some limitations of current methods for effort estimation by analogy. The accuracy and consistency of the derived estimate depends on the quality of the historical data and also on whether the method is able to find analogies between the historical projects and the one being estimated. In the first situation it is useful to calibrate the method to the local data while in the second case it is useful to utilize a tool that identifies projects that cannot be estimated with the classical EBA approach.

Recently, EBA has been improved significantly as a method [10] In this study however, EBA calibration is again done globally, without paying attention to potential estimation inaccuracies for specific projects.

In this study we will utilize EBA approach as implemented by BRACE tool [15]. BRACE tool has a tuning phase during which it determines the best parameters and the attribute subset that will participate in the estimation procedure based on certain accuracy statistics calculated during the estimation process of the historical data. We will further extend EBA method by:

a) Determining the situations under which the most critical configuration parameters of the method (i.e. the measures of similarity) are not appropriate for the estimation of particular projects.

b) Identifying a new set of parameters and attributes that are more appropriate for the estimation of the particular projects.

In particular data that are generated during the selection of the best parameters and attribute subset are analyzed in order to identify project attributes that lead to decreased estimation accuracy. Projects that present best configuration parameters different from the default ones are isolated in order to extract rules that identify more appropriate configuration parameters for the particular projects.

The proposed method is applied and evaluated in the widely known ISBSG data set release 7 [8]. The paper is organized as following: Sections 2, 3 and 4 involve the description of the methods, section 5 presents the data set used, in section 6 we present and discuss the results and in section 7 we conclude the paper.

2. Analogy Based Estimation

Analogy based estimation is essentially a form of case based reasoning. The basic aspect of the method is the utilization of historical information from completed projects with known size, effort or productivity. The most appropriate attributes are selected according to which the new project is compared with the old ones in the historical dataset. The attribute values are standardized (between 0 and 1) so that they have the same degree of influence and the method is immune to the choice of measurement units.

Initially, it is necessary to characterize the new active project, with attributes identical to the ones of the completed projects registered in the database. Examples of project attributes are the implemented functionality, programming language and application type. Attributes are distinguished as quantitative (such as function points [1], measuring the functionality of a software system) or qualitative [3] (such as programming language, measured in a nominal scale with values, “c”, “java” e.t.c).

The next step is to calculate how much the new project differs from the other projects in the available database. This can be done by using a «distance» metric between two projects, based on the values of the selected attributes for these projects. The most known distance metric is the Euclidean or straight-line distance which has a straightforward geometrical meaning as the distance of two points in the k-dimensional Euclidean space:

$$d_{new,i} = \left\{ \sum_{j=1}^{k} (Y_j - X_{ij})^2 \right\}^{1/2}, \qquad i = 1, 2, ..., n$$

Other possible distance metrics are the Minkowski distance, the Canberra distance, the Czekanowski coefficient and the Chebychev or «Maximum» distance (see [2] for definitions).

Eventually the estimation of the effort using analogies is based on the completed projects that are similar to the new one. The user of the method has to calculate the distances of the new project from all the database projects and identify few «neighbour» projects, i.e. those with relatively small distance value. The estimation of the effort is eventually obtained by some combination of the efforts of the neighbor projects. Typically, the statistic used is the mean or the median of these effort values. This statistic may be weighted according to the size (physical or functional) of the system under development compared to the sizes of the neighbor projects.

The prediction accuracy of the method may differ according to the peculiarities of the historical data set to which the target project is compared. For this reason, it is a good practice to calibrate the method, i.e. find out empirically the combination of method parameter values that provide the best accuracy results. The method options that may be adjusted are:

(a) The distance metric by which the projects of the database will be sorted according to their similarity to the one under estimation (e.g. Euclidean distance, Manhattan distance)

(b) The number of closest projects (analogies) – it is reasonable to expect that for small data sets a low number of neighbor projects must be used (typically one or two), while for larger data sets the choice depends on the homogeneity of the data,

(d) The statistic for productivity estimation

(e) The weighting of the chosen statistic according to the projects size

In this study we apply and calibrate analogy based estimation for size and effort prediction with the help of Brace tool [15]. The tool finds the best overall parameter configuration given a data set, by trying all feasible combinations of parameter values. Because of the computation complexity for large data sets, the user has to choose the range of values of certain parameters, deciding in advance values for some parameters based on his intuition and experience, in order to reduce the search space. Other researchers follow the same approach as well [13].

3. Rule Induction

Rule Induction [7] is a particular aspect of inductive learning. Inductive learning is the process of acquiring general concepts from specific examples. By analyzing many examples, it may be possible to derive a general concept that defines the production conditions.

Rule induction takes each class separately and tries to cover all examples in that class, at the same time excluding examples not in the class. This is a so called, covering approach, because at each stage a rule is determined that covers some of the examples. Covering algorithms operate by adding tests to the rule that is under construction, always trying to create a rule with maximum accuracy. Unlike other algorithms that choose an attribute to maximize the separation between the classes (using information gain criterion), the covering algorithm chooses an attribute-value pair to maximize the probability of the desired classification.

In this paper, we apply the PART algorithm that is based on extracting partial decision trees utilizing the Weka machine learning library [16].

A simple rule coming from the domain of software cost estimation will have as Rule Body certain software project attribute values and as a Rule Head a productivity (or cost, or effort) value. A simple example of a rule is the following:

If language used = java and development type= enhancement then 40<productivity ≤60 total no of projects= 10 wrong estimates=2

.

This rule is interpreted as following: If the language that will be used for the development of new project is java and the development type of the project is enhancement then there is (10-2)/10 = 80% (confidence value) probability that the productivity value of the project will be between 40 and 60 lines of code per hour.

One advantage of inductive learning over other machine learning methods is that the rules are transparent and therefore can be read and understood. Proponents of RI argue that this helps the estimator understand the predictions made by systems of this type.

4. Proposed Method

The proposed method involves initially the application of analogy based estimation for the identification of the best similarity measures using jackknife method. In each turn, one project is drawn out of the historical data set and used for estimation purposes. For this project we test a variety of combinations of similarity measures and we select the combination that presents the best MMRE estimation accuracy. MMRE is the Mean Magnitude Relative Error and is defined as following:

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\left| \frac{P_i - \hat{P}_i}{P_i} \right|$$

P_i, is the actual productivity required for the completion of a new project i, $\hat{P}_i$ the estimated productivity of anew project and n is the number of projects in the data set.

This procedure is repeated until all projects are excluded once and estimated by the rest of the projects. Finally the combination of similarity measures that presents the best overall estimation accuracy measured with MMRE is selected as the best configuration.

At this point we have determined a way to select similarity measures in order to apply EBA and generate an estimate for a certain unknown project. However, there maybe exceptions in which the new unknown projects present few or no similarities at

all to the historical data based on the selected measures. In such situations it is useful to have an enhancement of a method that specifies such projects and suggests alternative similarity measures.

For this purpose we further analyze the data produced for each project separately by jackknife method. We keep meta-data relative to the estimation procedure of each project separately, containing information about the most accurate distance metric, statistic point estimate and number of analogies for each project. The next step is to compare the accuracy of the estimations produced using the default similarity measures identified globally from the previous step to the accuracy of the best configuration identified for each project separately. Then we analyze the above data in order to produce rules that will indicate project attribute combinations that may lead to different best configuration.

5. Data set description

The data set used in the study is ISBSG release 7 [8], a publicly available multi organizational data set. The International Software Benchmarking Standards Group maintains a repository of international software project metrics to help developers with project estimation and benchmarking. The repository contains 1239 projects that cover the software development industry from 1989 to 2001.

Table 1. Data set description

Variable	Name	Levels values
Development Type	DT	1={enhancement, re – development} 2={new development}
Development Platform	DP	1={MainFrame} 2={PC,MidRange}
Language Type	LT	1={3GL, ApG} 2={4GL}
Programming Language	PPL	1={access} 2={cobol, cobolII, easytrieve, visual basic, natural, other4GL, otherApG, PL/I, powerbuilder, talon}
Database	DBMS	1={acess}, 2={ims} 3={adabas, db2, db2v2, foxpro, idms, sql, oracle, other, watcom}
Used Methodology	UM	1={yes} 2={no}
Organization Type	OT	1={Banking, ElectricityGasWater} 2={communication, community services, computers, defense, energy, financial, government, medical, professional services, wholesail&retail trade}
Business area type	BAT	1={accounting,banking,engineering} 2={r&d,claims processing, financial, insurance, inventory, legal, personnel, s&m, telecommunications}
Application type	AT	1={MIS, advertising, corporate taxation, data warehouse, DSS} 2={Transaction Processing System, Office Information System}
Package Customization	PC	1={Don't Know, Yes} 2={No}

In many records a number of fields are empty or even measured with different approaches. Our target was to include in the study the majority of the projects but also to ensure data validity minimizing the variance between the data because of the differences in measurement, or quality, two conflicting targets. The preparation and

transformation of data performed involved the selection of projects with data quality rating A and B (projects with data quality rating C were excluded). Projects for which only the development team effort and support was counted and only staff hours were recorded were selected. At this point 556 projects are considered but if we delete the cases with missing values, the dataset is restricted to 52 projects. Most of the predictor variables are categorical and since the building of a reliable model requires the existence of enough observations in every interaction of the values between dependent and independent variables, we chose to work with fewer categories. This approach is adopted in a similar study [11] where for each one of these categorical variables, one-way ANOVA was performed in order to check the impact of every factor on the original-dependent variable. Every factor with significance less than 0.05 was considered important and was included in the analysis. In [11] the authors used also post hoc tests (Tukey, Tukey's-b, Bonferroni, LSD, Scheffe and Duncan) in order to identify the various homogeneous categories that have to be concatenated in every factor. The categories that are not significantly different can be concatenated. Table 1, presents the variables that participate in the study along with their possible values.

6. Results

As mentioned above, we chose 52 projects characterized by the attributes presented in table 1. From these 52 projects the ten most recent ones implemented after 1997 were used as a validation set while the rest 42 projects were used to train the EBA method. Initially the BRACE tool was used to identify the most appropriate estimation parameters based on the projects in the training set. The parameters for which the tool decided upon are the distance metric, the number of analogies, the statistic used and whether the statistic will be adjusted based on the size of the project.

Figure 1 presents the alternative parameters that can be used for analogy based estimation, among which the tool specifies the optimum ones. The selection of the distance metrics by which the projects of the database will be sorted according to their similarity to the one under estimation is performed comparing 7 possible metrics, the Euclidean, the Manhattan, the Minkowski, the Canberra, the Czekanowski, the Chebychev and the Kaufmann-Rousseeuw distance metrics. The number of neighbor projects from the historical set that will be used for the estimation of a new project are limited from 1 to 5. The upper limit of 5 is selected based on the general assumption that for small data sets the number of analogies used should be around three [3].

The next choice is the statistic used to calculate a point estimate derived. A closely related decision with that choice is whether the statistic will be size adjusted, based on a size attribute. This statistic is actually used to calculate a point estimate based on the effort interval predicted. Possible statistics that the method compares is the mean point, the median point, the size adjusted mean and the size adjusted median. Another choice that can be made at this point is the selection of the project attributes that will participate in the estimation. In our case we select the participation of all attributes in identifying neighbor projects. The combination of parameter values that maximize the overall accuracy of analogy based estimation in the whole training set is the one selected as the best configuration of the model. The best parameters for EBA in the ISBSG data subset considered are presented in figure 2.

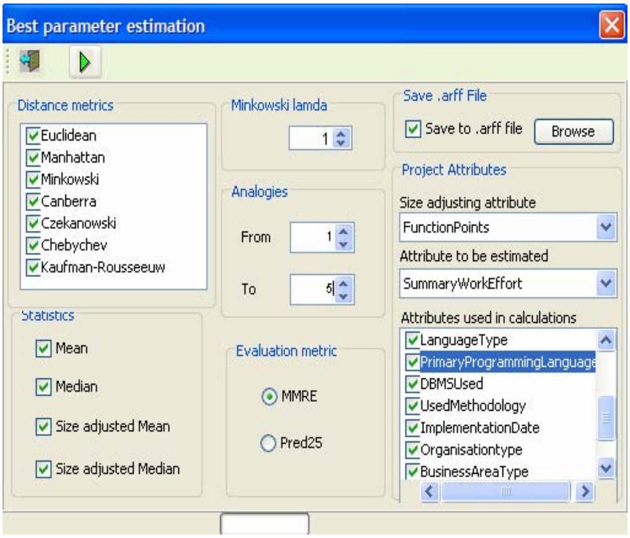


Figure 1: Similarity measures that participate in the configuration of ABE

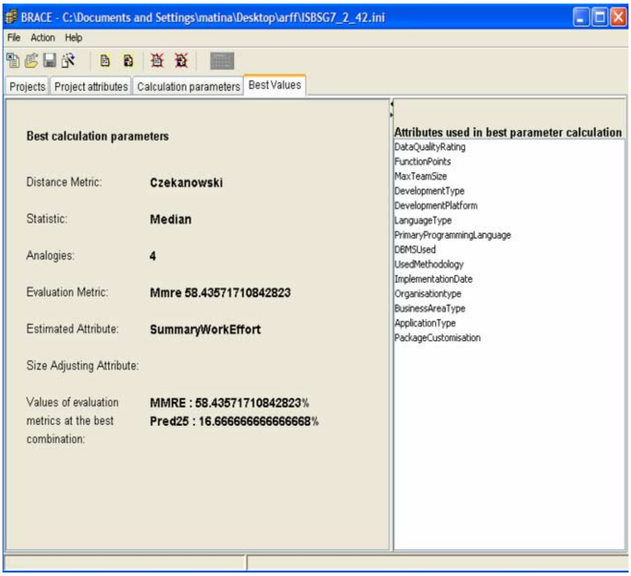


Figure 2: Best overall configuration of EBA method for ISBSG data set.

The optimal EBA based on ISBSG data takes into consideration the values of 4 neighbor projects measuring similarity using Czekanowski distance metric and provides point estimates using the median point of the interval without size adjustment. Using this configuration the overall fitting accuracy of the model to the training data evaluated with MMRE (Mean Magnitude Relative Error) metric is 58.43% and with Pred(25) (percentage of projects that are estimated with error less than 25%) is 16.66%. Such metrics are rather modest, since a combination of MMRE less than 25% and Pred(25) greater than 75% is considered quite satisfactory.

While searching for the best global configuration, the values of the best combination of parameters have been recorded for each project in a separate file. This file contains information regarding the best distance metric, number of analogies and statistic for each project alone. Using this information we extract rules that based on the projects attributes suggest the selection of a particular a) distance metric, b) statistic c) number of analogies.

The tool used to extract such models is the Weka Machine Learning Library. The files that are generated by BRACE tool can be readily processed by Weka.

Table 2. Rule set for the estimation of distance metric

PPL=1	Euclidean (8.0)
OT = 2	Czekanowski (4.0/2.0)
MTS <= 2 *and* LT = 1	Czekanowski (3.0)
DBMS =2 *and* FP <= 124 *and* LT = 2	Minkowski (4.0)
DBMS= 2AND FP <= 385 *and* LT = 1 *and* MTS <= 5	Euclidean (3.0/1.0)
LT = 1 *and* FP <= 385: Minkowski	Minkowski (13.0/3.0)
LT = 2	Euclidean (4.0/1.0)
	Chebychev (3.0/1.0)

Table 3 Rule set for the estimation of the statistic

PC=2 *and* BAT=2	Size adjusted Mean (11.0/4.0)
DP=1 *and* PC=2 *and* MTS > 7	Size adjusted Mean (7.0/1.0)
DP=1 *and* AT=2 *and* FP > 109	Mean (9.0/4.0)
DP=1 *and* AT = OTHER	Size adjusted Median (5.0/2.0)
DP=1 *and* LT=1	Size adjusted Mean (4.0/1.0)
FP <= 174	Size adjusted Median (3.0/1.0)
	Median (3.0/1.0)

Table 4. Rule set for the estimation of the number of analogies.

DT = 2 *and* OT = 2 *and* DBMS = 1	2 (6.0/4.0)
DT = 2 *and* OT = 1 *and* AT = 2 *and* FP <= 247	4 (16.0/8.0)
DT=2 *and* OT = 1 *and* AT = 1	3 (8.0/4.0)
DT = 1	2 (5.0/1.0)
OT = 1	5 (4.0/1.0)
	1 (3.0)

The Weka tool is further utilized to provide rules that will help the estimator select the best configuration for the estimation of a particular project based on the values of its attributes. The algorithm used is PART with the default parameters apart from the minimum number of projects per class which is set to 3. Tables 2,3, and 4 present the rules extracted for deciding the distance metric, the number of analogies and the statistic.

For example the first rule of table 2 suggests the use of Euclidean distance when the project under estimation is implemented in a “Programming Language” that belongs to the first category (MS Access). The particular rule suggests the Euclidean distance metric for 8 projects in the training set. For all of these projects the Euclidean distance is the optimal distance metric.

In the rule set that estimates the statistic metric the rule that classifies most of the projects is the first one that suggests the use of the size adjusted mean when PC belongs to the second PC category (see Table 1) and the BAT variable takes values

from the second BAT category. This rule suggests the optimal statistic for 7 out of 11 projects which are classified by this rule.

The estimation of effort for these projects based on the optimal overall configuration and the configuration suggested by the rules is presented in table 5. We should mention that three projects from the test set were removed because they were considered as outliers (IDs 25965, 29418 and 19037) because of their unusually small values of effort that did not appear in the training set.

The accuracy of ABE using the overall configuration and the accuracy of ABE with the configuration suggested by the rules is presented in table 6. Results show that there is significant improvement in the accuracy of the models, and that accuracy metrics are much closer to be considered satisfactory.

Table 5. Estimation of effort with the two configurations

Project ID	Actual effort	Global configuration	Rule based configuration
19165	893	961	719
23229	591	528	643,4
31085	170	546	115, 4
25965	21	546	84,1
15199	183	546	174,8
18268	212	732,5	353,6
29418	17	546	155,9
19037	30	423	176,5
23358	391	528	258,7
112583	1464	1038,5	1438,2

Table 6. Accuracy metrics for the two configurations.

	EBA (global configuration)	EBA (rule configuration)
Mmre	106,77	23,91
pred(25)	28,57	57,14

7. Conclusions

In this paper we have suggested a new approach to configure the method of EBA based on the historical data set used for the evaluation. During the training of the model in order to find the best overall configuration information about the best individual project configuration is maintained. This information is further analyzed in order to produce rules that will try to estimate the most appropriate distance metric and number of analogies and statistic for each project.

The results indicate an improvement in terms of accuracy of the proposed method compared to the default configuration. The method, unlikely global configuration, is able to identify for each project separately a particular set of similarity measures values that can identify the most analogous projects in the historical data set. The proposed method takes into consideration the unique attributes of each project deciding upon the distance metric, the number of analogies and the statistic providing a non-parametric model. The new approach increases estimation accuracy and reliability regarding the measures MMRE, Pred25 and provides another way of calibrating the method to the local data.

Future work involves the extraction of rules that will estimate, based on the projects attributes, the attribute subset that is best to participate in identifying neighbor

projects. The method looks promising and should be further evaluated in larger data sets.

8. References

[1] Allan J. Albrecht, John E. Gaffney Jr.: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. IEEE Trans. Software Eng. 9(6): 639-648 (1983)

[2] L. Angelis, I. Stamelos. A Simulation Tool for Efficient Analogy Based Cost Estimation, Empirical Software Engineering, Kluwer Academic Publishers, Vol 5 (1) pp. 35-68, March 2000

[3] Angelis, I. Stamelos, M. Morisio: "Building a Software Cost Estimation Model Based on Categorical Data", Proc. 8th IEEE International Symposium on Software Metrics, pp. 4-15 (2001).

[4] S. Bibi, I. Stamelos, L. Angelis: Software Productivity estimation based on Association Rules, in the proceedings of the European Software Process Improvement Conference, 13 A.6, Trondheim, Norway, November 2004.

[5] B. Boehm. Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ (1981).

[6] Nan-Hsing Chiu, Sun-Jen Huang: The adjusted analogy-based software effort estimation based on similarity distances. Journal of Systems and Software 80(4): 628-640 (2007)]

[7] D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, MIT Press, 2001

[8] International Software Benchmarking Standards Group, http://www.isbsg.org

[9] R. Jeffery, I. Wieczorek, A comparative study of cost modeling techniques using public domain multi-organizational and company-specific data, Proceedings of ESCOM-SCOPE 2000, Munich Germany, 18-20 Apr. 2000, Shaker Publishing B.V, pp239-248.

[10] Li, J., Ruhe, G., Al-Emran, A., and Richter, M. M. 2007. A flexible method for software effort estimation by analogy. Empirical Softw. Engg. 12, 1 (Feb. 2007), 65-106.

[11] P. Sentas, L. Angelis, I. Stamelos: Software Productivity and Effort Prediction with Ordinal Regression, Journal of Information & Software Technology, 47 (1): 17-29, (2005).

[12] M. Shepperd, C. Schofield, B. Kitchenham: Effort Estimation Using Analogy. ICSE 1996: 170-178

[13] Martin Shepperd, Chris Schofield, "Estimating Software Project Effort Using Analogies," IEEE Transactions on Software Engineering, vol. 23, no. 12, pp.736-743, 1997.

[14] Stamelos, L. Angelis, M. Morisio, E. Sakellaris, G. Bleris, Estimating the development cost of custom software, Information and Management 40 (2003) 729-741.

[15] Stamelos, L. Angelis, E. Sakellaris. BRACE: BootstRap based Analogy Cost Estimation, Proc. 12th European Software Control Metrics, pp. 17-23 (2001)

[16] Witten, I., and Frank, E. "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-327

Knowledge Acquisition/Transformation to Meet the Quality Needs of the Metallic Constructions Market

Nicholas P. BLESSEOS[1] and Athanassios F. BATZIAS
Department of Industrial Management & Technology, University of Piraeus, Greece

Abstract. Small/Medium Enterprizes (SMEs) of the industrial subsector of metallic constructions suffer of lack of means and knowledge to apply quality and process control. This paper deals with the design of a Technical Support Center (TSC), capable to provide the required knowledge to SMEs, by using as a knowledge vehicle the remedial advice offered by the TSC when defects are observed in the products without any evidence about the causing fault in the production line. For this purpose, a methodological framework has been developed under the form of an algorithmic procedure, including 26 activity stages and 8 decision nodes. Fault Tree Analysis (FTA) in its fuzzy version (to deal with uncertainty) was adopted as an inference engine for knowledge extraction and know-how transfer. An implementation referring to the anodization of aluminium is presented, with emphasis on cause-of-fault investigation/identification and communication protocols functioning. The relation of this procedure with (i) a complex/modified Nonaka model and (ii) 2nd order cybernetics is discussed while a university spin-off is suggested as mostly suitable to play the role of STC, inasmuch as this view is further supported through experience accumulated into the Laboratory of Simulation of Industrial processes at our Department.

Keywords. knowledge acquisition, diagnostic knowledge base, fault tree analysis, knowhow transfer, Nonaka model, 2nd order cybernetics, spin-off

Introduction

The processes met in most industrial units providing metallic materials/elements to the constructions sector of the economy are casting, extrusion, formation by pressing/cutting/welding/bolting/reveting/hammering/die-forging and surface treatment, including nitride hardening (nitriding), phosphorylation, anodization/passivation of aluminium and electroplating of steel. The great majority of the firms involved in this business consists of Small/Medium Enterprizes (SMEs). Even when electrochemical reactions are involved, which give, by their nature, very precise results, the variance of control parameters which is usually high under industrial conditions, implies faults at processing level and defects at product level. Some of these defects are not observable but will possibly influence the function of the corresponding article during its lifecycle. Most SMEs have not the proper equipment and the required knowhow in advanced quality control in order to (a) diagnose/identify/remedy a significant number of kinds

[1] Corresponding Author: Department of Industrial Management & Technology, University of Piraeus, Karaoli & Dimitriou 80, 185 34 Piraeus, Greece; E-mails: nblesseos@unipi.gr; fbatzi@unipi.gr

of defects, (b) optimize production conditions and (c) establish/maintain a Diagnostic Knowledge Base (DKB) that will facilitate technology transfer/adaptation. On the other hand, there are certain metallic construction departments operating within large companies possibly engaged, directly or indirectly, with national defense (e.g. there are two such companies in Greece). These departments are not competing the SMEs, because their production is dedicated to serve the main purpose of the large enterprise to which they belong. They also (a) follow medium/high quality standards, (b) engage personnel with increased qualifications and (c) operate well equipped laboratories for quality control; nevertheless, a great deal of knowledge remains either tacit or dispersed in thousands of official routine papers, completely unstructured and hardly retrievable.

The aim of this work is to develop a methodology for creating/maintaining a DKB for knowledge acquisition/transformation to meet the quality needs of the metallic constructions market, which can be used for (a) knowledge management either in the metallic construction Department of a Large Company (DLC) or in the operation of a Technical Support Center (TSC) and (b) technology transfer towards a SME in a mode of routine consultation on a quasi online / real-time basis. Such a TSC can be established either in a bottom-up way by the interested SMEs or in a top-down way by (i) a public Organization/Institution, (ii) an applied research center, (iii) a university, (iv) a DLC allowed to offer services to externals.

In Section 1, we present the methodology designed/developed/adopted, putting emphasis on knowledge management in both parts, the SME and the TSC, as well as in their communication/interaction, from a cybernetic point of view. In Section 2, we present an implementation of this methodology on surface treatment (anodization) of aluminium, putting emphasis on consultancy to be) provided by the TSC to the SME, using fuzzy FTA as an inference engine and experimental simulation for testing and making remedial proposals. In Section 3, we discuss the material presented herein and the relevant knowledge we have acquired and accumulated as intellectual capital in the Laboratory of Simulation of Industrial Processes and the research Group of Systems Analysis, both operating/performing in the Department of Industrial Management and Technology at the University of Piraeus, Greece. The discussion is concentrated on (i) the knowledge acquisition mechanism through the TSC/SE interaction as represented by a modified Nonaka's SECI model, (ii) the impact of the involvment of human factor, by referring to 2nd order cybernetics, and (iii) the possibility of establishing a university spin-off to play the role of a TSC addressing to SMEs in the industrial sub-sector of the metallic constructions (mainly anodization of aluminium and electroplating of steel). In Section 4, we conclude with (i) managerial implications for improving knowledge diffusion mechanisms and (ii) suggestions/proposals for further research in knowledge acquisition through the interaction between product quality and industrial process control.

1. Methodology

The methodology adopted herein, under the form of an algorithmic procedure, relies heavily on fuzzy reasoning within a top-down/bottom-up dialectic scheme of Fault Tree Analysis (FTA), to deal with uncertainty, and includes the following 26 activity stages and 8 decision nodes (see Figure 1 for their interconnection):

1. Setup of the network including the TSC and the SMEs under consideration.
2. Setup of the communication protocols.
3. Observation of a fault in articles produced by the SME.
4. Examination of production conditions.
5. Remedial proposal, confirmation and corrective action.
6. Complete description of the fault in an internal report.
7. Preliminary identification of the fault via the DKB.
8. Recording of the production conditions, referring to the lot in which the defected article was observed.
9. Coding of message according to MF1 [variables/conditions/documents (standard photos, diagrams) reported under strict specifications]; this Message Format (MF) is obligatory in the response/remedial – proposal sent by the consulting/ supporting TSC to the supported SME.
10. Coding of message according to MF2 [Message Format allowing for exposing the problem in a narrative way (accompanied by available documentation, e.g. non-standard photos), because a strictly prescribed form cannot apply in the situation under consideration].
11. Coding of message according to MF3 [Message Format, that simply accompany defected articles or guide specimens (set purposely at preselected sites in the electrolytic bath) when they are dispatched to the consulting TSC].
12. Electronic transmission of message to TSC, according to the established communication protocols in stage 2.
13. Conventional dispatch of message to TSC (defected article or part of it or representative sample of the defected material is included).
14. Measurements in the Quality Control Laboratory, according to mutually accepted standards and recommended practices.
15. Delivery decoding, and (if necessary) interpretation of message (to assign contextual meaning when description in natural language is used for giving complementary information), according to the principles of 2^{nd} order Cybernetics.
16. Knowledge-based identification of fault.
17. Computer aided localization of the corresponding fault tree where the identified defect is the top event.
18. Successive causes path identification through the assignment of fuzzy significance values on events by an expert (or group of experts participating in a Delphi method to investigate the causal chain most likely responsible for the fault appearance).
19. DKB Creation/enrichment receiving information internally (see Figure 1) and externally via an intelligent agent, like the one described in [1].
20. Experimental confirmation (after scaling down) of the corrective action.
21. Updating of the fuzzy rules used in the fault tree, if necessary.
22. Conditional Proposal for correction action to the production department.
23. Unconditional Proposal for correction action to the production department.
24. Coding of response (feedback) message, according to MF1, and electronic transmission to SME.
25. Decoding of the message by the SME and application of the corrective proposal (given by the TSC), after finding the correspondence to production conditions (proper interpretation, according to 2^{nd} order Cybernetics).
26. Registration in the SME's local KB and the TSC's central DKB.

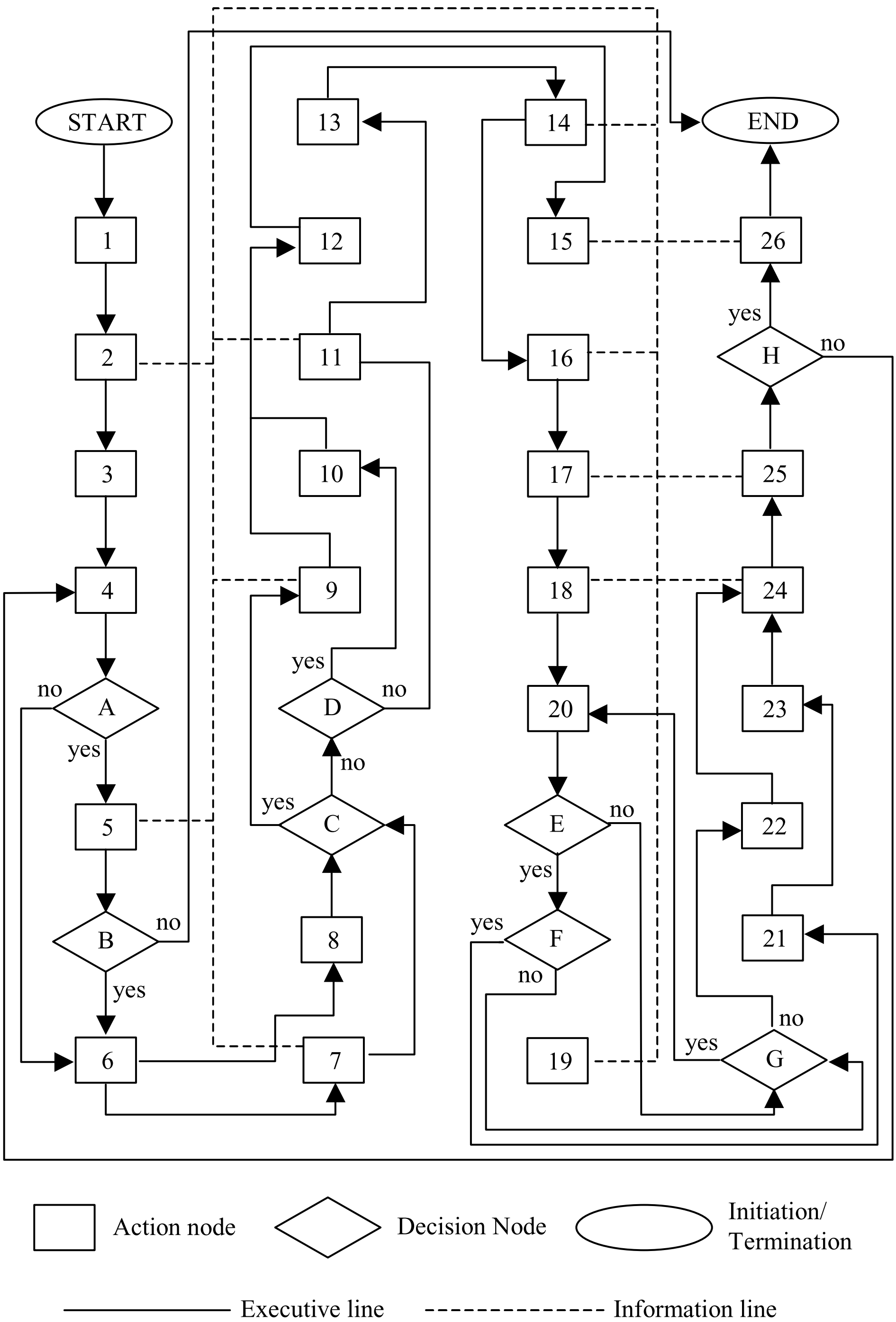


Figure 1. The methodological framework, under the form of an algorithmic procedure, developed for interactive knowledge acquisition/processing and know-how transfer from a SME to a TSC.

A. Is there any parameter or control variable or a combination of them out of the recommended interval or combination of intervals?
B. Is there any defected article in the new lot?
C. Is the preliminary identification complete?
D. Is the unidentified defect describable under the specifications required by the MF2 terms?
E. Is the confirmation of the path successful?
F. Is the remedial proposal testing feasible?
G. Is there another experimental test?
H. Is the application successful?

Obviously, a Rules Based Reasoning (RBR) approach has been adopted as a background for the algorithmic procedure described above. Since, in practice, an interplay of rules and cases is unavoidable, all cases examined, either actually proved successful or unsuccessful, are stored to be used as raw material for Case Based Reasoning (CBR) when either an unidentifiable defect appears or an holistic rearrangement of rules is needed.

2. Implementation

The methodology described above has been successfully implemented in the case of electrochemical anodizing of aluminum plates. Emphasis was put on the RBR approach in contrast to a recent publication of our Research Group [2], where fuzzy multicriteria choice among alternatives obtained by FTA, was used to provide the proposed solution. The latter method was proved useful only in the cases where a SME had already an adequately equipped laboratory and the required know-how for quality/ process control at a minimum level capable to provide the necessary information for multicriteria analysis (which might be incorporated herein as a new dendritic extension of stage 18) to be performed together with experimental testing by a TSC.

Most of the necessary information was obtained from the Hellenic Aerospace Industry SA and the Laboratory of Simulation of Industrial Process of the Department of Industrial Management at the University of Piraeus, Greece. The case example presented herein refers to a severe defect described in a narrative way (see screenshot in Figure 2), following the communication protocol MF2 (stage 10), since there is not corresponding standard terminology according to the respective fault tree (already in the DKB), and its vocabulary. By simulating the TSC behavior, the system localizes, according to stage 17, as most relevant the defect ‘uneven anodic coating’ under the numerical code 1.3.2 (see Table 1 and Figure 1A in the Appendix) which is set a top event for FTA. Applying the rules on input fuzzy data, we obtain the result on the top event in fuzzy form, which can be defuzzified to give a crisp number that is more familiar to the operator in the SME. In Figure 3, an extract of a specimen run is shown, starting from the leaf of the tree ‘insufficient agitation’, which is experimentally found the mostly responsible ultimate cause, and ending at the top event. The linguistic variables refer to each event, considered in the universe of discourse of the corresponding fault, and appear as indices (%), with terms L,M,H (Low, Medium, High, respectively, according to a three-point Likert Scale). The rules are set in the usual IF - THEN form; e.g., IF 1.3.2.2.3.1 is L AND 1.3.2.2.3.2 is M THEN 1.3.2.2.3 is M.

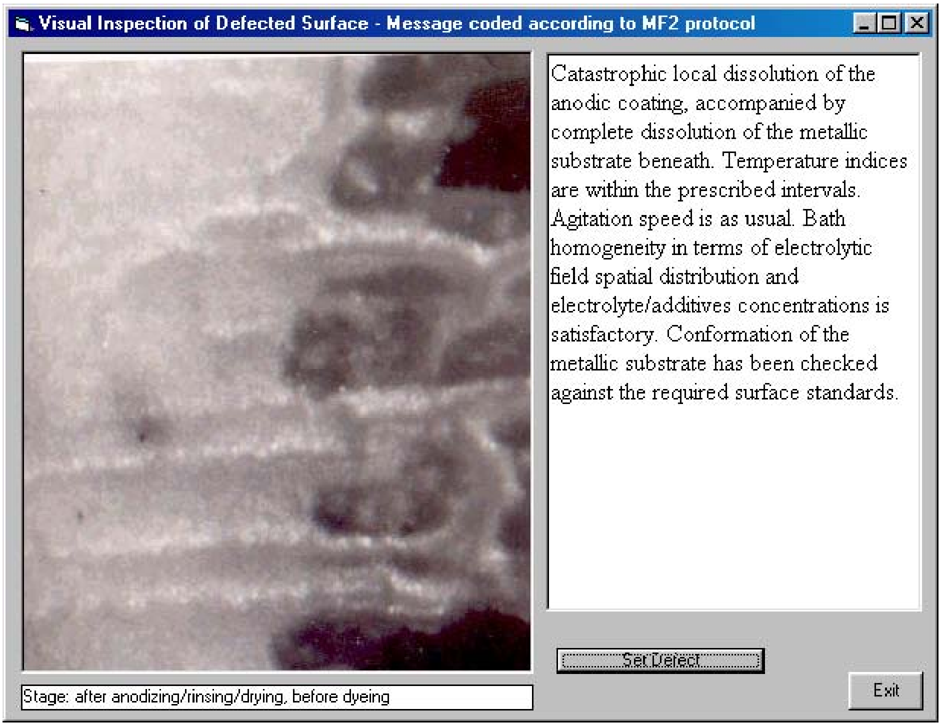


Figure 2. Screenshot indicating electronic message from SME to TSC, coded according to MF2 protocol, referring to the local dissolution of anodized aluminum.

Table 1. Intermediate/Final events in the partial tree under examination.

1.3.2	Uneven anodic coating
1.3.2.1	Uneven distribution of electrolytic field
1.3.2.2	Inhomogeneity of bath
1.3.2.2.1	Non-uniform distribution of electrolyte concentration
1.3.2.2.1.1	Insufficient agitation
1.3.2.2.1.2	Bath ageing
1.3.2.2.2	Non uniform distribution of additives concentration
1.3.2.2.2.1	Insufficient agitation
1.3.2.2.2.2	Bath depletion of additives
1.3.2.2.3	Non uniform distribution of temperature
1.3.2.2.3.1	Insufficient agitation
1.3.2.2.3.2	Uneven cooling
1.3.2.3	Inhomogeneity of metallic surface
1.3.2.3.1	Insufficient etching
1.3.2.3.2	Insufficient desmudging
1.3.2.3.3	Appearance of new oxides
1.3.2.3.3.1	Formation of natural aluminium oxide
1.3.2.3.3.2	Oxide deposits as result of local corrosion
1.3.2.3.4	Insufficient rinsing
1.3.2.3.5	Dirt, grease and finger-marks caused during handling before anodizing
1.3.2.4	Inhomogeneity of compact film

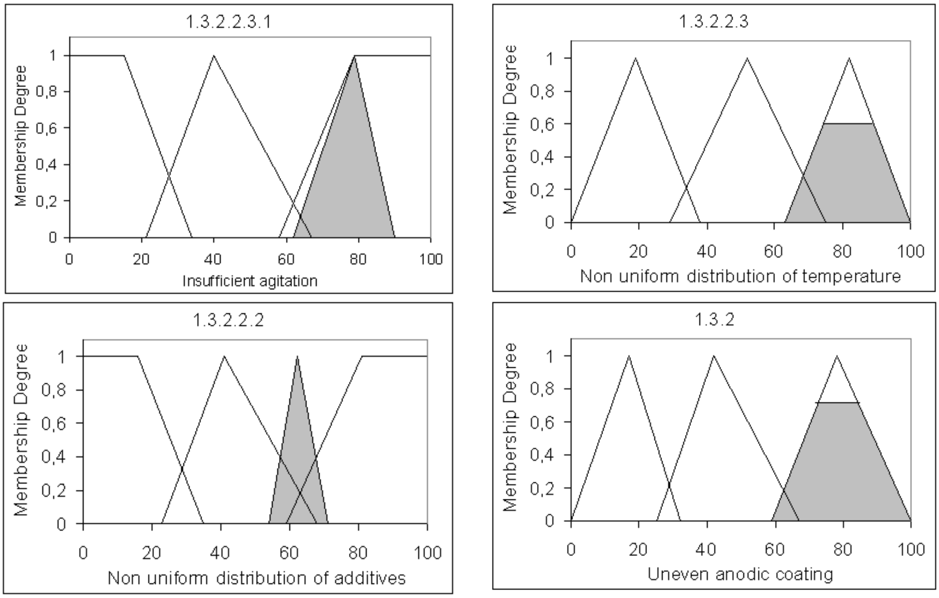


Figure 3. A sample of fuzzy input and intermediate/final output (left and right hand schemes, respectively) representing FTA for the branch marked in gray in Figure A1 in the Appendix.

3. Discussion

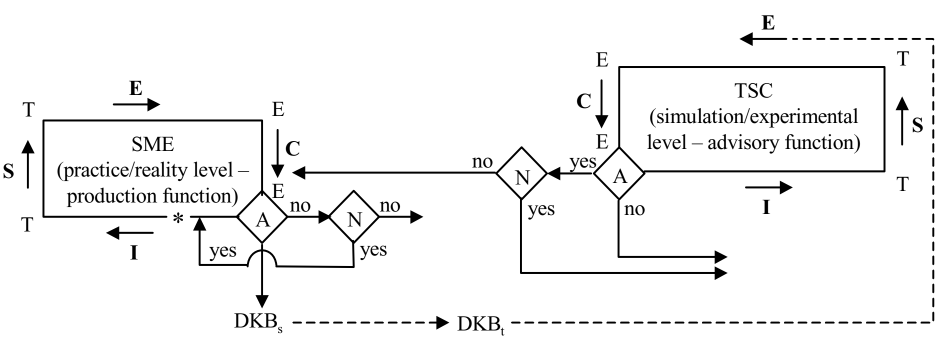
From a conceptual point of view, we can represent knowledge acquisition/processing/ conversion, as described in Section 1 and shown in Fig 1, by using the Nonaka's SECI model [3]: Socialization (S), for converting freshly acquired tacit knowledge through communication/sharing/diffusion at experiential level. Externalization (E), for articulating tacit into explicit knowledge either for hypotheses making, to be subsequently tested under real or simulated conditions, or for simple prototypes preparation, to be subsequently examined for further improvement. Combination (C), for converting simple/modular knowledge into more complex and systematic sets of explicit knowledge, including hypotheses testing and dissemination/diffusion at formal/ agreed level. Internalization (I), for embodying explicit into tacit knowledge by the individuals involved, mainly through learning by doing/observing and sharing formal knowledge.

Our modification of Nonaka's model consists in coupling two knowledge subsystems, one for SME and another for the TSC. These subsystems are shown in Figure 4 where the asterisk signifies the defect observation as information given explicitly, the symbols on the arrows stand for the SECI model while, the symbols E, T on each apex stand for explicit and tacit/implicit knowledge, respectively. The local DKB of each SME, named DKB_s, and the central DKB which belongs to TSC, named DKB_t, are interconnected for information exchange.

In terms of cybernetics, TSC is an observer/agent while SME is observed entity, both integrated within a system where interaction between them takes place. Since the

'observer' as a cybernetic subsystem tries to construct a model for another cybernetic subsystem (i.e. the 'observed'), we have a "cybernetics of cybernetics", i.e. a 'meta' or 2nd order cybernetics. As a matter of fact, we have an analog to the Principle of Uncertainty as extracted from quantum mechanics, which states that observer and observed cannot be separated, and the result of observation will depend on their interaction. This effect is not immediately obvious to the engineer in charge for quality management in the SME, since he knows thoroughly the internal structure of the production line to a high degree of accuracy as well as the points where a fault may appear; therefore, he tends to de-emphasize the system/model distinction, thinking as if the model is actually the subsystem under consideration; such an engineer behaving as a 1st order cyberneticist, tends to manage the subsystem under his supervision as if it were a passive, objectively given 'thing', that can be separately observed and independently managed. In the contrary, a 2nd order cyberneticist recognizes the same subsystem (herein, the SME) as an agent in its own right, interacting with another agent, herein the TSE), since limited knowledge or/and malfunctioning of the latter agent may lead to a vicious circle through this interaction (due to bilateral feedback mechanism, possibly as a result of misinterpretation of the decoded proofs/descriptions quoted in stages 15, 25 of the algorithmic procedure shown in Fig. 1), a circulatory problem arises. A solution to this problem might be a permanent link of the TSC with a knowledge provider, functioning in a different mode; such a link is the relation of an Organization of public interest (established by the state, the Local Authorities or a Professional Union) with a spin-off having the Organization as a parent.

On the grounds of the above analysis, we can propose a university spin-off (USO), company as a feasible solution to the problem of establishing an independent/effective/ reliable and widely accepted TSC. The main advantage of such a company over a corporate spin-off (CSO) comes from the difference in intellectual or mental character of the people running these spin-off firms, who carry the spirit of the antecedent organization: while a private company often tries to keep research and technology within the firm selling solely products/services in a black-box form, a university usually allows or even encourages the transfer of results to outsiders in more or less transparent form. Moreover, because of the large dissimilarities between the parent organizations, it seems natural that the continued relation between the USO and the



A : Has a satisfactory remedy been achieved?
N : Is continuation of cyclic SECI procedures in the same subsystem recommended?

Figure 4. Interconnected subsystems forming a complex/modified Nonaka model, representing knowledge acquisition/processing/transformation; symbols in bold stand for processes quoted in the SECI model.

University, compared to the CSD and the private firm, is expected to be completely different. E.g., a CSO can easily be an associate customer or subcontractor to its parent firm (possibly developing a common attitude as regards the extent/degree up to which knowledge diffusion is allowable), while this is almost impossible for an USO. On the other hand, a CSO may develop a competitive character its parent organization, consequently eliminating a path of knowledge transfer, a situation absolutely impossible in the case of a USO, since its activities are completely different in comparison to those of the parent University.

4. Conclusions and Remarks for Further Research

A methodological framework, under the form of an algorithmic procedure, based on fuzzy fault tree analysis, has been developed for interactive knowledge acquisition/processing and know-how transfer from a metallic constructions SME to a TSC. This procedure was applied successfully in the case of electrochemical anodizing of aluminum, using data provided by the Hellenic Aerospace Industry SA and the Laboratory of Simulation of Industrial Process of the Department of Industrial Management at the University of Piraeus, Greece. The conceptual background of the methodological framework was shown to follow a complex/modified Nonaka SECI model while the interaction between SME and TSC can be (i) studied through 2nd order cybernetics and (ii) satisfactorily realized by means of a university spin-off.

Suggestions for further research include (i) extension of FTA to consider human errors expected to occur in the interaction between industrial operator and electrochemical process equipment by analogy with the interaction between a computer user and the operating system, according to [4], and (ii) incorporation of the TSC's consultancy into the SME's managerial planning system, developed in [5] as a knowledge-based approach.

Acknowledgements

Data supply from the electrochemical processing departments of the Hellenic Aerospace Industry SA and financial support provided by the Research Centre of the Piraeus University are kindly acknowledged.

References

[1] F.A. Batzias, E.C. Marcoulaki, Restructuring the keywords interface to enhance CAPE knowledge via an Intelligent Agent, Computer-Aided Chemical Engineering 10 (2002), 829 – 834.

[2] A.F. Batzias, Fuzzy multicriteria ranking of aluminium coating methods. *Computation in Modern Science and Engineering*, Vol. 2, Part B, AIP Conference Proceedings, New York, 2007, pp. 856-861.

[3] I. Nonaka, R. Toyama, N. Konno, SECI, *Ba* and leadership: a unified model of dynamic knowledge creation, *Long Range Planning* **33** (2000), 5-34.

[4] M. Virvou, Automatic reasoning and help about human errors in using an operating system, *Interacting with Computers* **11** (1999), 545-573.

[5] N.P. Blesseos, G.P. Chondrocoukis, A group decision support system design for small and medium enterprises. *Advances in production Management Systems*, Elsevier Science Publishers BV. The Netherlands, 1991.

Appendix

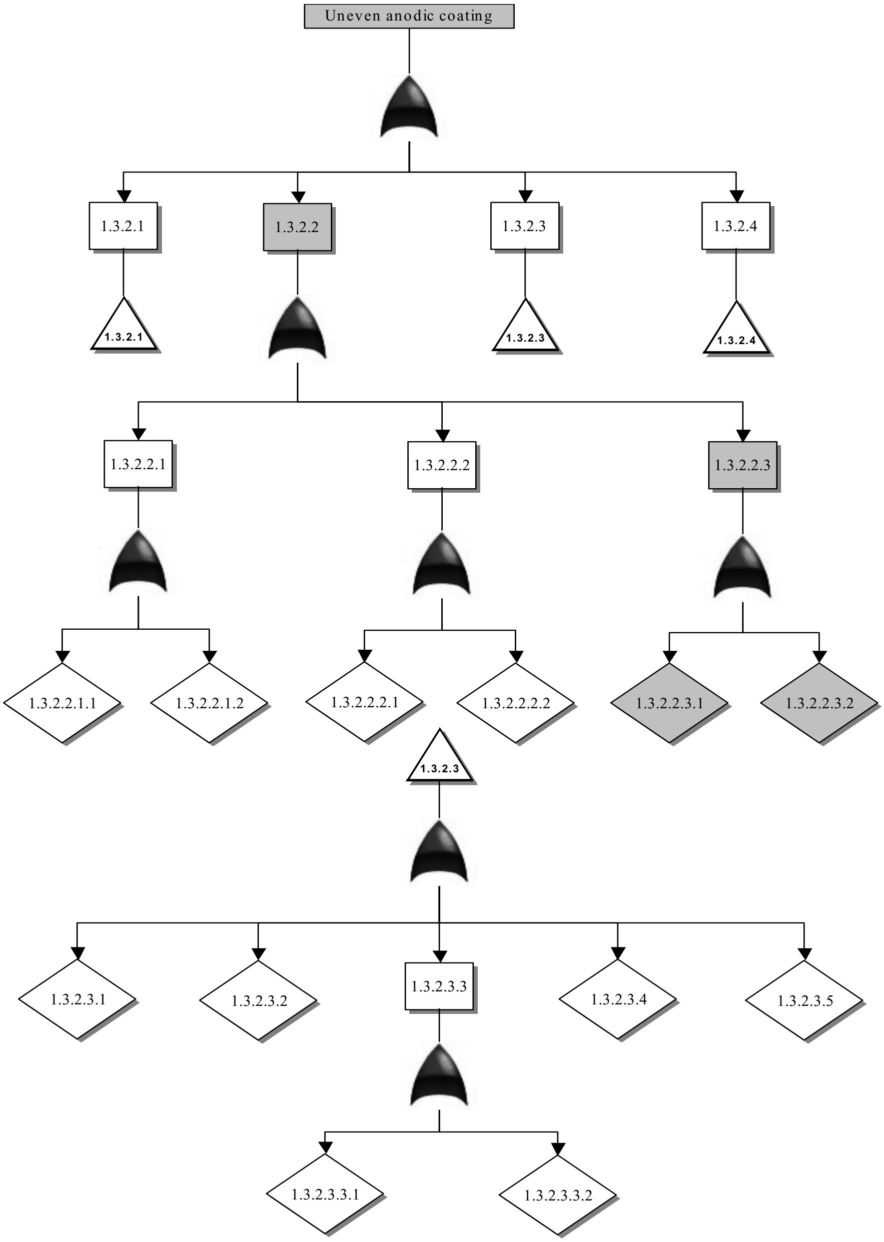


Figure A1. Fault tree analysis, when the defect considered as top event is ‘Uneven anodic coating’.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-337

Knowledge Acquisition within a Network of Industrial Laboratories Performing Quality Specifications Testing

Panayiotis FOTILAS and Athanassios F. BATZIAS[1]
Department of Industrial Management and Technology, University of Piraeus, Greece

Abstract. This work deals with knowledge acquisition by an industrial small/medium enterprise (SME) using quality control services provided by external laboratories/agents, in order to establish functional relations between processing conditions and product properties. The laboratories are considered to constitute a formal Normal network or an informal Research network (N-NetLab or R-NetLab, respectively). A cognitive algorithm is designed/developed as an aid for suggesting the most proper cooperative schemes that might take place for the realization of this outsourcing procedure within both networks. As a case example, an enterprise producing aluminum-silicon casting alloys is examined by means of the suggested algorithmic procedure. Last, computer-aided (by means of case based reasoning – CBR) filtering-correction of information expressed in narrative mode is presented, using also contextual analysis within an ontological framework.

Keywords. Knowledge acquisition, quality control, Nonaka model, ASTM standards, implicit/tacit knowledge.

Introduction

An important problem, which Small/Medium Enterprizes (SMEs) face in quality control, is the lack of (i) adequate laboratory equipment and (ii) corresponding knowledge. The solution to this problem is (i) outsourcing through cooperation with reliable external agents, capable to offer such services under strict agreement specifications and (ii) creation/operation of a Knowledge Base (KB) within the enterprise, so that knowledge acquired in the time course can be stored in easily retrievable/usable form. The agents are either independent laboratories or technological centers established by SMEs-Unions providing technical support to the member of each Union or laboratories in departments of large enterprises, offering such services. In several cases, an agent cannot by itself satisfy the requirements of quality testing, especially if this testing should be realized under a standard specification or a recommended practice, as this is described by the corresponding document of a Standardization Documentation (ISO, ASTM, DIN, BS). As a result, a network (LabNet) of systematically or occasionally cooperating agents exists, which may have a formal name that facilitates membership in corresponding international bodies (e.g. in Greece there is such an association named HELLASLAB, which is a member of the

[1] Corresponding Author: Department of Industrial Management and Technology, University of Piraeus, Karaoli & Dimitriou 80, 18534, Piraeus, Greece. Email: fbatzi@unipi.gr

EUROLAB). For the needs of the present study, we name it Normal LabNet (N-LabNet), because we consider also the set of laboratories that belong to universities, research institutes, and applied research centers, characterizing it as Research-LabNet (R-LabNet); the members of this set communicate to each other by means of conferences, publications, symposia, colloquia, workshops, and sometimes undertake projects to solve quality control problems, when these are of scientific interest.

The present work deals with knowledge acquisition by an industrial SME through quality examination of an intermediate/final product carried out mainly by the aid of the N-LabNet. The goal is to identify the relation between product properties/qualities and production conditions in order to enable correction action in the process when a defect appears in the product. Since, in this case, knowledge acquisition is realized after experimental confirmation of a hypothesis (about a possible relation) formed by SME's experts on the basis of (i) explicit input, under the form of data-information, and (ii) their subjective/implicit/tacit knowledge, we have adopted the Nonaka's SECI model [1], which deals with similar concepts and conversion processes, although in a somehow different context. According to this model, there are four stages in knowledge creation: Socialization (S), as the process of converting shared experience into new tacit knowledge; Combination (C), as the process of converting rather simple explicit to more complex explicit knowledge; Internalization (I), as the process of converting explicit into tacit knowledge, by embodying the former into the experts' cognitive background. The repetition of these four stages, which constitute a quasi cyclic procedure, if considering a conceptual projection level, forms a spiral directed towards phenomenological levels of higher cognitive content and information granularity. Since in the present work the combination stage should be replaced by a confirmation stage, testing the hypothesis formed at the previous E-stage, we are actually in front of the classic inductive-deductive scheme of scientific reasoning as regards knowledge creation: assimilation (I_o), internal processing at lower/basic semi-conscious/experiential/subjective/individual level (S_o), hypothesis formation (E_o), confirmation by experimentation/observation (C_o), I_1, S_1, E_1, C_1,…, C_n, where C_n stands for final confirmation after *n* repetitions. To the extent that the initial stage of this knowledge-creation mechanism starts by processing an input given usually in explicit form (i.e. the first stage is I), it should be better to use the name ISEC, adding also a symbol for the node that makes the decision on whether confirmation of hypothesis will take place at the next (upwards) phenomenological level of the spiral.

1. Methodology

The methodological framework we have developed, under the form of an algorithmic procedure, for knowledge acquisition within a network of industrial laboratories performing quality specifications testing, includes the stages described below. Figure 1 illustrates the interconnection of the stages, represented by the corresponding number or letter, in case of activity or decision node, respectively.

1. Selection/evaluation of data/information/knowledge concerning production conditions and control variables/parameters and their relation to properties of intermediate/final products. (IF)
2. Application in small scale production.
3. Explicit relation proposal, under feasible technoeconomic constraints [2], and experimental design for confirmation/validation. (EF)

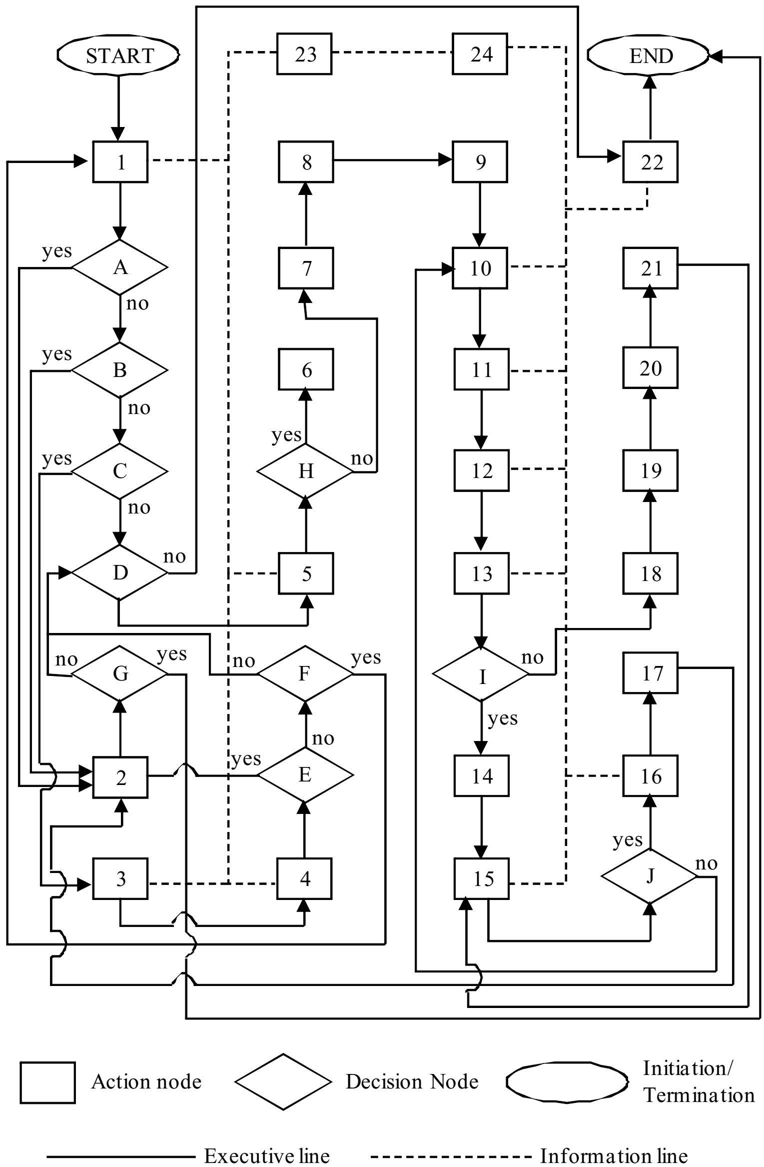


Figure 1. Flow chart of the algorithmic procedure designed/developed herein for suggesting optimal cooperative schemes within the N-LabNet and R-LabNet, including the KBs creation/enrichment/operation.

4. Performance of experiments in laboratory, scale up, and evaluation of results to set up a functional relation. (CF)
5. Collection of the agents most likely to be in position to undertake the project.
6. Formation of a set of agents applying such standard methods and/or practices.
7. Formation of a set of agents occupying the most relevant equipment and employing personnel with proper expertise.

8. Determination of criteria for choosing the agent closer to requirements of the project.
9. Multicriteria choice of agent.
10. Assignment of the project to the chosen agent, giving also the relevant data/information acquired so far (especially as regards the progress made in determining the relation described in stage 1) in explicit form; knowledge internalization by embodying explicit information into tacit knowledge. (IJ)
11. Communication among experts within the agent for exchanging relevant experience. (SJ)
12. Presentation of parts (including alternatives) that might constitute one or more versions of a relation satisfying preset requirements/constraints. (EJ)
13. Selection/modification/connection of parts to present an integrated entity in the form of a proposed relation (or most probably a set of proposed relations). (CJ)
14. Performance of the respective experimental work. (CJ)
15. Evaluation of results as regards the confirmation of proposed relations. (CJ)
16. Documentation and preparation of (i) guide text/information for operators/users and (ii) report to the enterprise.
17. Demonstration to the personnel of the enterprise.
18. Selection of agents capable to carry out the required experimental work.
19. Determination of criteria for choosing the most capable collaborator.
20. Multicriteria choice of collaborator.
21. Performance of the assigned experimental work. (CJ)
22. Search for locating the scientific topic closest to variables/parameters under consideration in order to suggest the corresponding ontology for attracting agents involved in R-LabNet.
23. KB operation/enrichment in the enterprise.
24. KB operation/enrichment in the agent, functioning in combination with a data mining mechanism to extract knowledge from external Bases [3].

A. Has the validity of such relations been confirmed within the same enterprise in the past? (IF)
B. Is there relevant data/information in technical literature and experience acquired by the personnel in the past, which might be used to establish such relations? (IF)
C. Is there a possibility (based on personnel's experience/knowledge) of suggesting such relations within the enterprise/organization? (SF)
D. Are there agents in the N-LabNet capable to undertake the project?
E. Are the results satisfactory on a quantitative, explicitly expressed basis?
F. Is the redesign feasible?
G. Is the application successful?
H. Are there agents applying relevant standard methods and/or practices?
I. Is the agent in position to confirm the proposed relation(s) by performing the respective experimental work?
J. Are the results satisfactory?

The criteria to be determined in stages 8 and 19 for choosing an agent/collaborator are similar: cost, time availability, expected time-to-respond, specialization of equipment and personnel, recent experience, reliability. Nevertheless, there are some differences: (i) the agent should be capable to contribute by offering original work to reveal the relation described in stage 1, therefore he should have also an adequate

scientific background, (ii) the collaborator should have significant relevant know-how mainly in practice.

2. Knowledge Acquisition

We can distinguish at least two SECI or ISEC cycles within which knowledge acquisition takes place. The first is inside the enterprise and includes the stages/nodes 1, 3, 4, A&B, which stand for I, S, E, C, respectively. The second is inside the agent (with a possibility of outsourcing the C-stage, depending on decision making through node I) and includes the stages/nodes 10, 11, 12, 13 (or 21, in case of outsourcing) &14&15, which stand for I, S, E, C, respectively. Since the C-stage refers rather to experimental confirmation than to simple combination (as defined within Nonaka's original model), it is always followed by a decision node with a question (as input) on the success of confirmation and three possible responds as output: (i) affirmative answer, which leads to 'application', (ii) negative answer, which leads either to agent or to failure (blind alley or dead end) if experimental testing is carried out by the enterprise or the agent, (nodes X and Y in Figure 2), respectively, and (iii) open answer, suggesting that, although success has not been achieved yet, repetition of the same cyclic procedure (at an upper level of knowledge, due to accumulation of new information created by means of 'learning by doing') may be proved crucial for obtaining a decisive affirmative or negative answer. The nodes deciding on repetition are symbolized with F and J in the algorithmic procedure shown in Figure 1, in correspondence with the nodes X and Y in Figure 2. Since each cognitive cycle, and the corresponding spiral (see Introduction), is characterized by a unique decision node, we can discriminate the stages of each cycle by combining the letter representing the Nonaka stage with the letter representing the decision node in the algorithmic procedure. These combinations are shown in parentheses quoted at the end of each stage described in the previous section in order to (i) facilitate correspondence between Figures 2 and 1 and (ii) highlight the knowledge acquisition mechanisms forming the backbone of the procedure developed herein.

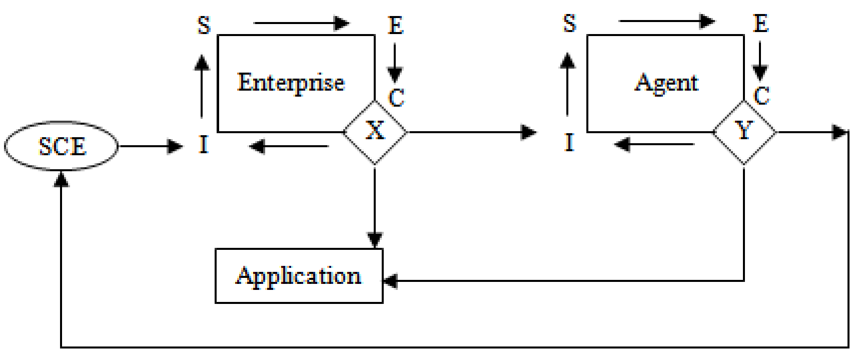


Figure 2. The cooperative scheme between Enterprise and the Agent/Laboratory chosen in stage 9 (see Figure 1); SCE stands for Start/Cont./End; the 'Application' activity refers to stage 2 of the algorithmic procedure shown in Figure 1.

3. Case Example

As a case example, we have examined an enterprise producing aluminum-silicon casting alloys. The most common and serious defect in this industrial product is porosity, which is a result of two possible faults: insufficient feeding and hydrogen entrapment during solidification. Since these defects cause costly scrap loss and limit the use of castings in critical, high-strength applications, the effect of alloying elements as well as the cooling rate on the amount of porosity retained in aluminum castings has to be quantified. Relevant data/information/knowledge can be obtained/acquired within the enterprise, according to stage 1, since the required equipment forms part of the quality control laboratory and the experimental design as well as the statistical processing of the results of performed measurements should be familiar to the personnel. More specifically, the required equipment includes induction furnace for melting the alloy ingots, rotary degasser, system of thermocouples to follow the temperature change, system to determine density of samples based on the fundamental Archimede's principle [4], spark emission spectrometer for chemical analysis of samples.

As regards the performance of experimental design, the execution of measurements and the evaluation of their results, according to stages 3 and 4, the use of Taguchi orthogonal arrays, followed by ANalysis Of VAriance (ANOVA) with interaction, is a recommended practice that can be carried out within the enterprise. The sand mold fabrication can be assigned to a sub-contractor, in which case this product follows also the procedure shown in Figure 1 (starting from Stage 1), with co-responsibility of the enterprise and the subcontractor, according to the clauses of the special agreement made for outsourcing. For testing the validity of results and revealing the mechanism underlying the empirical relation found by statistical analysis, in order to enable intervention for improving the product quality, scanning electron microscopy (SEM) as well as Energy Dispersive Xray (EDX) analysis is recommended to be performed by an agent (stages 6-17), in case that the quality laboratory of the enterprise is not properly equipped, which is a common situation in industrial practice. In case that the chosen agent covers only SEM examination, he should find a collaborator to carry out EDX analysis, according to stages 18-21 in case that there are not available agents in the N-LabNet, stage 22 can be activated through decision node D to search for locating the scientific topic closest to variables/parameters under consideration (which is casting within the scientific branch of Physical Metallurgy) in order to find out a proper laboratory in the informal R-LabNet.

4. Ontological Filtering/Correction of Information

Any kind of information expressed in narrative mode entering the KBs of either the enterprise or the agent (stages 23/24) undergoes testing for correctness / consistency through contextual examination as regards its meaning and formality. A semantic netwotk (like WordNet, where sets of synonyms, called synsets, and their semantic relations are found) within the corresponding ontology is used as a framework for such an examination. For this purpose, explicit knowledge is used for applying common semantic relations, like the following (referring to P): meronymy (P is part of Q); holonymy (P has Q as part of itself); hyponymy (P is subordinate of Q); hypernymy (P is subordinate of Q); synonymy (P denotes the same as Q); antonymy (P denotes the

opposite of Q). On the other hand, implicit/tacit knowledge is used by the experts at experiential level to judge by analogy (to similar cases examined in the past) and propose correction action, if necessary, as a matter of fact performing what is known (in the expert systems practice) as Case Based Reasoning (CBR).

An example, coming from the records of the Laboratory of Simulation of Industrial Processes of the Department of Industrial Management and Technology, at the University of Piraeus, Greece, is the following, referring to quality control of aluminum-alloy plates fabricated for pressure vessels and aerospace applications.

According to the standard practice B 594 – 06 of ASTM for ultrasonic inspection of aluminum-alloy wrought products for aerospace applications, this test "is employed to detect internal discontinuities oriented in a direction parallel to, or nearly parallel to, the surface of the product". Evidently, there is an infinite number of 'directions parallel to the surface'; therefore, this expression cannot enter the KB as is, because it cannot be used to guide the operator how to perform ultrasonic inspection. In practice, the operator probably knows by experience how to perform the test but the lack of commitment through a clearly expressed clause in the document of agreement between the enterprise and either the customer or the agent does not offer formal/legal coverage to the customer or the enterprise, respectively. Computer aided CBR is used by the expert to retrieve (actually, 'retrieve' is the first r-step in the four r-step CBR, the rest three being 'reuse', 'revise', 'retain') the analogue standard test method B 548 – 03 of ASTM for ultrasonic inspection of aluminum-alloy plate for pressure vessels, where it is quoted that this test "is employed to detect gross internal discontinuities oriented in direction parallel to the rolled surface". The new keyword that may promote computer aided association (instead of pure mental association) is 'rolled', which, in its turn, gives other synonym keywords within the synsets of the WordNet-KB in order to obtain finally the right phrases 'rolling lines', 'milling lines', 'extrusion lines', which must accompany the word 'surface' in the above mentioned extract as follows: "…parallel to the lines still apparent or 'memorized' (although eliminated by cleaning) of the surface …". The same procedure is followed when visual material is used as a complement to natural language description. The photos in Figure 3, extracted from the KB of our Laboratory, show the direction of apparent and 'memorized' extrusion lines, along which internal discontinuities such as cracks, raptures, laminations should be detected by means of the above mentioned standard methods/practices of ASTM.

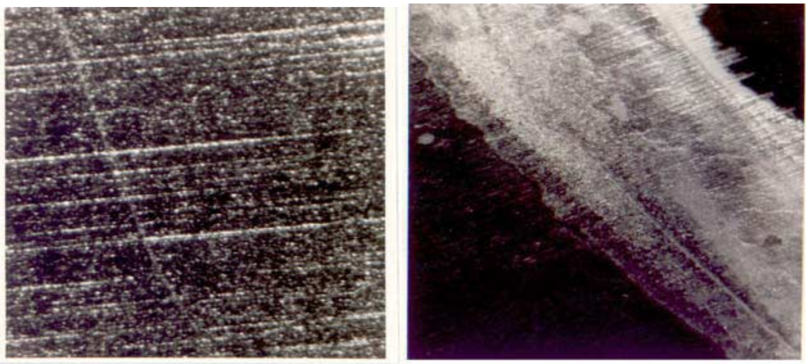
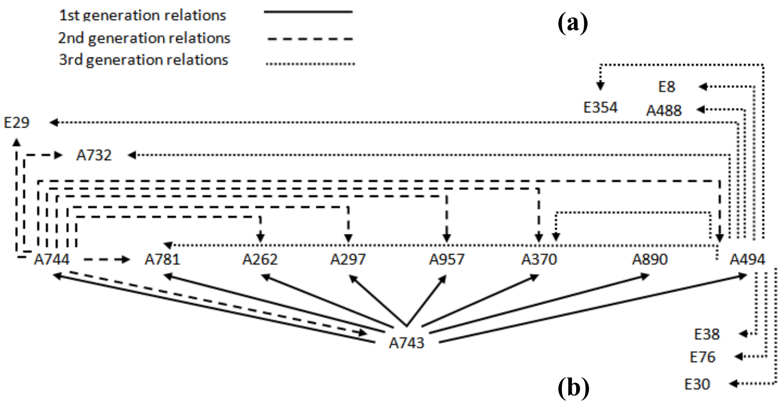


Figure 3. Traces of extrusion lines on anodized aluminum surface, as they appear under the microscope: (a) topography of final surface after normal anodization; (b) topography of surface after catastrophic anodic dissolution (due to high electric density causing local overheating), in which case the direction of lines is still evident, because the solid substrate 'remembers' the distortion that it has undergone during extrusion.



(b)

⟶	A743	A744	A781	A262	A297	A957	A370	A890	A494
A743	0	1	1	1	1	1	1	1	1
A744	1	0	1	1	1	1	1	0	1
A781	1	1	0	0	1	0	1	1	1
A262	0	0	0	0	0	0	1	0	0
A297	0	0	1	0	0	0	1	0	0
A987	0	0	0	0	0	0	0	0	0
A370	0	0	1	0	0	0	0	0	0
A890	0	0	1	0	0	0	1	0	0
A494	0	0	1	0	0	0	1	0	0

⟶	A744	A262	A297	A370	A494	A732	A743	A781	A957	E29
A744	0	1	1	1	1	1	1	1	1	1
A262	0	0	0	1	0	0	0	0	0	0
A297	0	0	0	1	0	0	0	1	0	0
A370	0	0	0	0	0	0	0	1	0	0
A494	0	0	0	1	0	0	0	1	0	0
A732	0	0	0	0	0	0	0	0	1	0
A743	1	1	1	1	1	0	0	1	1	0
A781	1	0	1	1	1	0	0	0	0	0
A957	0	0	1	1	1	1	1	0	0	0
E29	0	0	0	0	0	0	0	0	0	0

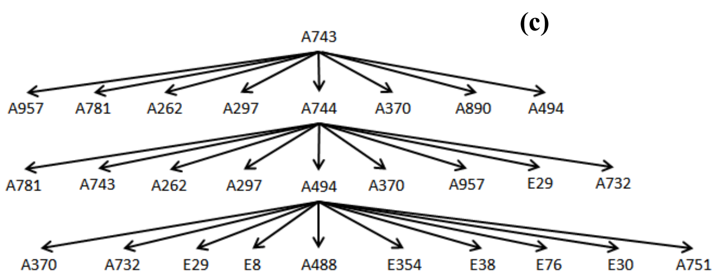


Figure 4. Representation of the knowledge path from explicit know-how in the industrial process of casting iron-chromium-nickel alloys (A743) to experiential/implicit knowledge acquired by personnel (A488): (a) as partial network; (b) as successive adjacency matrices; (c) as an inverse tree with repeatable leaves.

CBR can be enhanced by creating a network of standards as cognitive background. This can be easily achieved, since each standard has a set of references to other standards, named ‘referenced’ (quoted as ‘applicable’ in older editions) documents. Starting from a certain standard, arbitrarily selected, we can represent it as well as its references with points or vertices or nodes interconnected with arcs directed from the initial standard to its references. These references are all members of the first generation with the initial standard as unique parent. Each member of the first generation has referenced documents of its own, creating a second generation, and so on. Evidently, the initial standard as well as any member of the k generation may be a member of one or more generations after the next one; i.e., a member of the k ($k = 0, 1, 2, \ldots$) generation may also appear as a member of the j generation, provided that $j \geq k+2$. This condition leads to the form of a directed multigraph, where loops with one node (resulting to a pseudograph) are not allowed. Values, indicating dependence of applying a standard (especially of carrying out a standard test) on the previous application of another standard of the next generation, are assigned to each arc, so that a network is obtained from the directed multigraph. These values vary from 0, indicating no dependence, to 1, indicating full dependence.

In the application example of the previous page, the initial standard specification for castings, iron-chromium, iron-chromium-nickel, corrosion resistant, for general application A 743/A 743M – 06 has eight arcs leading to corresponding referenced documents of the first generation. By considering the simplest case of full dependence (value = 1, assigned to each arc), for sake of illustration, we can formulate the corresponding adjacency matrices as well as a tree like structure with repeatable leaves (see Figure 4), which may facilitate algorithmic programming for computer aided consultancy. In this illustration, the knowledge path from explicit know-how in the industrial process of casting iron-chromium-nickel alloys (A743) to experiential/implicit knowledge acquired by personnel (A488) is represented.

5. Discussion

The algorithmic procedure, described in Section 2 and shown in Figure 1, is a dynamic one, since it covers circumstances that may appear in the time course. E.g., if the assignment of the enterprise to a specialized in radiography agent is the examination of heated-treated carbon, low-alloy, and martensitic stainless steel castings and the agent finds indications of existence of discontinuities, then he has to search for (in agreement with the enterprise) a collaborator specialized in ultrasonic examination of such steel castings; this means that decision node I may be reactivated with a time lag, which could not be predicted at the time of initial assignment. Such a reactivation may be proved also necessary in case that the enterprise is willing to decrease uncertainty by cross-testing the product under examination; in the above mentioned example, if the agent follows the standard practice A 609/A 609M – 91 (reapproved 2002) of ASTM with calibration by using a series of test blocks containing flat bottom holes and feels uncertain about the result, he may ask a collaborator to check the results performing the alternative calibration technique by using a back wall reflection from a series of solid blocks.

On the other hand, the agent may serve as an expert, especially when surface inspection of the product by visual examination has been agreed between the enterprise

and the customer. E.g., if the agent examines steel castings according to the standard practice A 802/A 802M – 95 (reapproved 2006) of ASTM then he should check production in real time for compliance with acceptance levels by utilizing Steel Castings Research and Trade Association (SCRATA) graded reference comparators for the visual determinations of surface texture/roughness/discontinuities. This means that decision node D is not used for outsourcing but for hiring a quality expert on a permanent or temporary basis.

6. Concluding Remarks

The methodological framework we have developed, under the form of an algorithmic procedure (including 24 activity stages and 10 decision nodes), for knowledge acquisition within a network of industrial laboratories performing quality specifications testing, has been proved to be (i) capable to incorporate Nonaka's knowledge creation model, (ii) consistent with the presented case example, referring to an enterprise producing aluminum-silicon alloys, (iii) functional for ontological filtering/correction of information, and (iv) suitable for cooperating with a network of standards to offer an aid for CBR. In the last case, the knowledge path from explicit know-how in the industrial process of casting iron-chromium-nickel alloys to experiential/implicit knowledge acquired by personnel is represented. The same methodology might be used for knowledge acquisition in the case of spatially distributed systems, like wetlands [5,6] examined for their conformation with environmental standards by means of biosensors and processing of measurements in real time. Such an extension could be proven valuable in creating by induction a cognitive model applicable to new waterbodies entering as new entities an already established environmental network.

Acknowledgements

Financial support provided by the Research Centre of the Piraeus University is kindly acknowledged.

References

[1] I. Nonaka, R. Toyama, N. Konno, SECI, *Ba* and leadership: a unified model of dynamic knowledge creation, *Long Range Planning* **33** (2000), 5-34.

[2] P. Fotilas, A.F. Batzias, Synthesizing equivalence indices for the comparative evaluation of technoeconomic efficiency of industrial processes at the design/re-engineering level, *Computation in Modern Science and Engineering. AIP Conference Proceedings* **963** (2008), 884-889.

[3] F. A. Batzias and E. C. Marcoulaki, Restructuring the keywords interface to enhance CAPE knowledge via an intelligent agent, *Computer Aided Chemical Engineering* **10** (2002), 829-834.

[4] M. Dash, M. Makhlouf, Effect of key alloying elements on the feeding characteristics of aluminium-silicon casting alloys, *Journal of Light Metals* **1**(2001), 251-265.

[5] A.F. Batzias, A.P. Efthymiadou, C.G. Siontorou, A knowledge based system offering consultation for enhancing semi-natural wetland functionality, *Computation in Modern Science and Engineering. AIP Conference Proceedings* **963** (2008), 878-883.

[6] A.F. Batzias, C.G. Siontorou, A new scheme for biomonitoring heavy metal concentrations in semi-natural wetlands, *Journal of Hazardous Materials* (2008), in press.

Methodology and Tools for Knowledge Discovery and Data Mining

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-349

Method for Building Dictionaries to Extract Reputation Information

Hiroshi MARUYAMA[a, 1], Kennshi SUZUKI[a, 1], Taichi NAKAMURA[a, 1]
[a] *Tokyo University of Technology*

Abstract. This paper proposes the method of building dictionaries for extracting reputation information. We propose a method of building a highly credible dictionary from the staggering volume of information that customers post on Web pages. The principle of our proposed method is based on assigning a certainty factor which represents the probability that extracted words really do express a feature or evaluation expression and registering only words which have a certainty factor greater than a pre-set threshold value. As a result of experiment, the method of building a highly credible dictionary has been established.

Keywords. Reputation, Extract, Syntax Structure pattern, Certainty factor, Confidence factor

Introduction

In order for companies to make improvements to their commercial products or services, they lay great emphasis on knowing what end users require and in obtaining feedback on end users' feelings of dissatisfaction with their products as evidenced in CGM (Computer Generated Media) such as messages posted on bulletin boards, Weblog or SNS. Even if a message posted on a bulletin board is anonymous, the believability does not decrease [1]. Therefore it may be useful to interpret the requirements of a customer from CGM.

In order to extract information about a product's reputation from CGM, existing research studies [2][3] have proposed methods of collecting the relationship between a characteristic or feature of an article and the evaluation of that feature by a user, and include studies to extract words which express such an evaluation to categorize the reputation of products. However, these methods involve significant costs in promptly registering words that represent features of new products which are being developed day by day in a dictionary, due to manual effort involved in building such a dictionary. It is generally difficult to extract the reputation information from the Web documents in CGM in a manner which guarantees both credibility and coverage.

In order to solve this problem we have developed a method of automatically expanding dictionaries by using the co-occurrence of a feature expression and an evaluation expression to alternately extract a word expressing a feature of an article and a word expressing an evaluation of that feature from Web documents written in

[1] Corresponding Author: Hiroshi Maruyama, Graduate School of Bionics, Computer and Media Sciences, Tokyo University of Technology, 1404-1 Katakura-Cho, Hachioji-Shi, Tokyo 192-0982, Japan; E-mail:hiroshi_maruyama@mf.teu.ac.jp

Japanese [4]. Although the dictionaries could initially have a high credibility, as the dictionaries were expanded we found we could not satisfy both high credibility and coverage, even when very common words were discarded on the basis of the TF-IDF weight. One of the reasons for the low credibility problem is that the extracted word may not necessarily describe a feature or the evaluation of a feature.

We propose a method of building a highly credible dictionary from the staggering volume of information that customers post on Web pages. The principle of our proposed method is based on assigning a certainty factor which represents the probability that extracted words really do express a feature or evaluation expression and registering only words which have a certainty factor greater than a pre-set threshold value.

1. Related studies

Research works relating to the automatic analysis of opinion information collected from the Web include the following topics: unsupervised learning algorithms for classifying reviews as “thumbs up” or “thumbs down” [5] and [6]; a method for quantitatively representing the degree of satisfaction with commercial goods and the degree of confidence in the results when analyzing opinion-related sentences [7], and a method for extracting a customer’s opinion from a sentence in which a character string includes a subject, and a method of building dictionaries that contain the feature and evaluation expressions [8].

The studies on using evaluation expressions extracted from opinion information collected from the Web have succeeded in classifying review documents or sentences as “thumbs up” or “thumbs down”. The evaluation expression might not be dependent on the particular field that the product relates to and may be useful for obtaining general comments about a product. However, the businessman thinking about the new idea of a product demands evaluations for the function of the product.

Tateishi *et al.* classify user's message according to the feature of the product, and they show a subjective factor with quantitatively that to influence the purchasing action of the end user, using two values, namely the satisfaction rating, which is the ratio of the number of positive messages to the total number of messages for the classification, and the certainty of the satisfaction rating, which is expressed by the number of messages for the classification [9]. Kobayashi *et al.* define an opinion in a document as consisting of an object, an attribute, and an evaluation expression, and propose a method of building a dictionary for an each expression and for every domain type [2]. They have released an evaluation expression dictionary that has 5,200 words registered manually [10].

2. Methodology for building dictionaries

2.1. Principle of methodology for building the dictionary

The customer feedback information for a product or a service consists of the name of the product, a feature expression, and an evaluation expression. A significant number

of words concerning the features of a product or service arise from day-to-day discussions in the market in which the new product or service has been developed. It is not practicable to make a general-purpose dictionary which contains all the feature expressions relating to all subject fields.

Our proposed methodology covers Web pages, or pages from word-of-mouth communication bulletin boards describing a relevant product, and extracts the user reputation information only in the form of a feature expression and an evaluation expression. We cannot analyze all the Web documents in the world, but we might be able to get enough information to ascertain customer opinions.

The volume of Web documents is increasing dramatically. More than several million pages (ex. iPod Shuffle : 12,200,000 hits by Google) may often be included in an analysis and it would consume too much time to build the dictionaries manually. According to our previous experiment the number of manually extracted words expressing a feature or an evaluation varies depending on the person who selected the words as expression words.

To solve these problems the dictionaries have to be built automatically. The proposed procedure for building the dictionaries is as follows: first define the various syntax structures in which a feature expression and an evaluation expression co-occur, then define a feature expression which obviously represents a key feature of a relevant product, and finally extract an evaluation expression which co-occurs with this feature expression in one of the defined syntax structures.

However it is not necessarily the case that the word in the structure extracted by using a co-occurrence relation between a feature expression and an evaluation expression does actually represent an evaluation. In other words the word extracted in this manner is characterized by the probability that it is an appropriate evaluation expression. This probability is named the certainty factor. The certainty factor is associated with and appended to the word which forms the evaluation expression. Then only words which have a certainty factor greater than the pre-set threshold value are registered in the evaluation dictionary.

Then in order to expand the feature dictionary new feature expression words are extracted based on the co-occurrence in the syntax structures with the evaluation expression words registered by the previous operation. In a similar way the newly identified feature expression words are registered in the feature dictionary. The two dictionaries can be expanded by successively and alternately repeating the two updating operations.

2.2. A syntax structure pattern and the certainty factor of a word

According to the results of our pilot study, the ratio of the number of appropriate reputation expressions to the total number of them extracted varied depending on the type of syntax structure pattern. This ratio is named the confidence factor of the syntax structure pattern. The confidence factor varies depending on the part of speech of the extraction word. Although there are many types of syntax structure which express reputation information, the following three typical types of syntax structures are used to represent almost all reputation expressions. Ga and Wa are small words (particles) used in Japanese to mark the subject and topic of a sentence.

Type 1. [feature expression]+ Ga (in Japanese) +[evaluation expression],
for example, "DEZAIN Ga YOI" means that design is good.

Type 2. [feature expression]+ Wa (in Japanese)+ [evaluation expression],
for example, "MIYASUSA Wa JYUUYOU" means that visibility is important.

Type 3. [evaluation expression]+ [feature expression]
for example, "YASUI KAKAKU" means low price.

The procedure for obtaining the confidence factor $c(PE_i)$ of a syntax structure pattern PE_i, which is used in extracting the evaluation expressions is as follows:

(1) Prepare feature expressions represent those features of the product in question which are obviously relevant. The set of these expressions is registered in the feature dictionary. The dictionary created in this manner is the initial feature expression dictionary.

(2) Collect Web documents by using the initial feature expression dictionary.

(3) Extract sentences corresponding with the syntax structure pattern PE_i referred to above from the collected documents.

(4) The word that appears at the position of the evaluation expression really demands the probability that is evaluation expression. Handle this according to a part of speech of the evaluation expression.

For example, in case that "design (DEZAIN in Japanese)" is registered in the initial feature dictionary as an obviously appropriate feature expression, extracted predicative adjective with type 1 syntax structure pattern (PE_1) has 10 words (ex. "YOI" in Japanese means good) and all of them were appropriate evaluation expression. In this case the confidence factor of $PE_1(c(PE_1))$ with predicative adjective is 1.0. The value of the confidence factor ranges from 0 to 1.

In a same way, the confidence factor $c(PF_i)$ of a syntax stricter pattern PF_j for extracting a feature expression can be obtained.

The procedure for building the initial evaluation dictionary is as follows:

(5) Evaluation expression word W_p is extracted from sentences including PE_i. Assign a value of $c(PE_i)$ to $e(W_p)$[2]. If the same evaluation expression word W_p appears in other syntax structure pattern, $e(W_p)$ is obtained by taking the geometrical mean of the confidence factor of other syntax structure patterns.

(6) Register these words and their associated certainty factor in the initial evaluation dictionary.

After two initial dictionaries have been built, the features expression dictionary is expanded as follows:

(7) Collect documents using the initial evaluation expression dictionary,

[2] A certainty factor of $e(W_p)$ is the probability that the evaluation expression word W_p which the system extracted is appropriate. In this case, a certainty factor of feature expression is 1.0, so the probability depends on the type of syntax structure pattern.

(8) Extract a sentence corresponding with the syntax structure pattern PFj which is used to extract the feature expression from the collected documents,

(9) Obtain the certainty factor $f(W_q)$ of a word W_q which appears at the position of the feature expression in the sentence including the syntax structure pattern PF_i, by multiplying the certainty factor $e(W_p)$ of the simultaneously occurring evaluation expression in the selected sentences and the confidence factor $c(PF_i)$ of the syntax structure pattern PF_i.

If the same feature expression word W_q appear in another syntax structure pattern m, the certainty factor $f(W_q)$ is taken to be the geometrical mean of the individual value. If the words are already registered in the features expression dictionary, keep a former certainty factor of them.

(10) Only words which have the certainty factor greater than the pre-set threshold value, τ, are registered in the features expression dictionary.

2.3. The expansion of dictionaries

Figure 1 shows the procedures for expanding the two dictionaries. The evaluation words which have been newly extracted by using the initial feature dictionary and the syntax structure patterns are added to the initial evaluation dictionary with the associated certainty factor. The feature dictionary is then expanded by adding words newly extracted by co-occurrence with words already registered in the evaluation dictionary. The certainty factor of the word which is already registered in the initial feature dictionary is not updated. The evaluation expression dictionary is further expanded by adding an evaluation word newly extracted by co-occurrence with a word already registered in the feature dictionary.

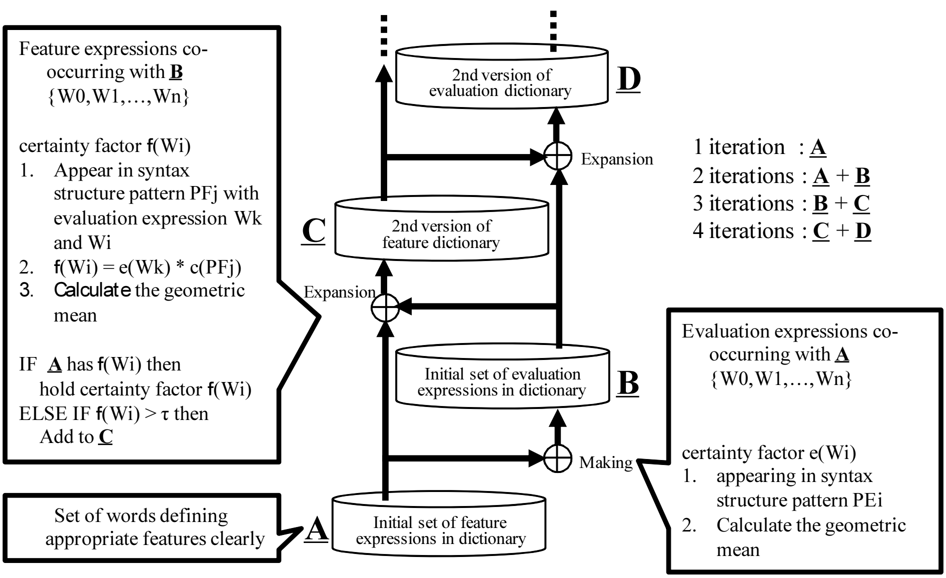


Figure 1. Step by step updating of the two dictionaries

3. Experiments

3.1. Experiment method

3.1.1. Documents collected

In order to show the effectiveness of the proposed method in being able to automatically build the dictionaries in a manner which achieves a high precision ratio, we first used a crawler to collect the documents of Japanese Weblogs concerning mobile telephones, which are an extremely popular product in Japan. 77,568 pages, with a total volume of 2.3GB, were collected over a period of about 45 days from November 12th 2007 to December 25th 2007. The documents which could be analyzed comprised 38,904 pages which were extracted from HTML documents. Spam documents were eliminated by keyword filtering techniques.

3.1.2. Initial features dictionary

Since a cellular phone is used daily by everyone, we assume that a name of a function implemented in a cellular phone is clearly established as a feature expression in anyone's judgment and words which express the name of a function have "KINOU" as a suffix. Words with the suffix "KINOU" were extracted from the 38,904 pages of usable documents and the any occurrences of words which included a grammatical error in the extracted sentences were eliminated. Finally 41 words which obviously represent the features of a cellular phone were registered in the initial feature dictionary.

3.2. Experimental results

Table 1 and Table 2 show the confidence factors $c(PE_i)$ and $c(PF_i)$ which were obtained from the 38,904 pages of usable documents for every syntax structure pattern. The syntax structure patterns consist of several elements, which we have labeled #0, #1 and #2. The item labeled #0 in the tables is the part of speech of the extracted expression, #2 is the anchor word, that is the known word already in the dictionary and used to identify the sentences from which words are extracted, and #1 is "Ga" or "Wa".

Consider the sentence of "DEZAIN Ga YOI" meaning "the design is good.". The syntax structure pattern of this example sentence is shown in the first column of Table 1. The predicative adjective, "YOI", which appears in this syntax structure pattern in position #0 is evidently an evaluation expression. In other words the confidence factor c(PEj) of the predicative adjective appearing in the syntax structure pattern shown in the first column of Table 1 is 100%.

Table.1 The confidence factor of the syntax structure pattern used to extract evaluation expressions

Syntax structure pattern	confidence factor
#2[feature], #1GA, #0[evaluation](predicative adjective)	1.000
#2[feature], #1GA, #0[evaluation](intransitive verb)	0.442
#2[feature], #1GA, #0[evaluation](general compound noun)	0.857
#2[feature], #1GA, #0[evaluation](noun - adjectival noun stem of a word)	1.000

#2[feature], #1GA, #0[evaluation](noun - adverbial)	1.000
#2[feature], #1HA, #0[evaluation](predicative adjective)	1.000
#2[feature], #1HA, #0[evaluation](intransitive verb)	0.208
#2[feature], #1HA, #0[evaluation](Noun - SAHEN connection)	0.286
#2[feature], #1HA, #0[evaluation](general compound noun)	0.250
#2[feature], #1HA, #0[evaluation](noun - adjectival noun stem of a word)	0.750
#0[evaluation](predicative adjective), #2[feature]	1.000
#0[evaluation](intransitive verb), #2[feature]	0.947
#0[evaluation](intransitive verb , auxiliary verb), #2[feature]	0.462
#0[evaluation](intransitive verb, intransitive verb, auxiliary verb), #2[feature]	1.000
#0[evaluation](Noun - SAHEN connection , noun – suffix - adjectival noun stem of a word | auxiliary verb), #2[feature]	1.000
#0[evaluation](general compound noun | noun – suffix - adjectival noun stem of a word | auxiliary verb), #2[feature]	1.000
#0[evaluation](noun - adjectival noun stem of a word | auxiliary verb), #2[feature]	1.000

Table.2 The confidence factor of the syntax structure pattern used to extract feature expressions

Syntax structure pattern	confidence factor
#0[feature](predicative adjective ,noun - suffix), #1GA, #2[evaluation]	1.000
#0[feature](intransitive verb), #1GA, #2[evaluation]	1.000
#0[feature](Noun - SAHEN connection), #1GA, #2[evaluation]	1.000
#0[feature](Noun - SAHEN connection ,noun - suffix), #1GA, #2[evaluation]	1.000
#0[feature](general compound noun), #1GA, #2[evaluation]	0.852
#0[feature](general compound noun , Noun - SAHEN connection), #1GA, #2[evaluation]	1.000
#0[feature](general compound noun , Noun - SAHEN connection), #1GA, #2[evaluation]	1.000
#0[feature](prefix – noun connection , Noun - SAHEN connection), #1HA, #2[evaluation]	1.000
#0[feature](prefix – noun connection, general compound noun), #1HA, #2[evaluation]	1.000
#0[feature](Noun - SAHEN connection), #1HA, #2[evaluation]	0.100
#0[feature](general compound noun), #1HA, #2[evaluation]	0.500
#0[feature](general compound noun ,noun – suffix – SAHEN connection), #1HA, #2[evaluation]	1.000
#2[evaluation], #0[feature](Noun - SAHEN connection)	0.500
#2[evaluation], #0[feature](Noun - SAHEN connection ,noun - suffix)	0.500
#2[evaluation], #0[feature](general compound noun)	0.341
#2[evaluation], #0[feature](general compound noun ,noun - suffix),	1.000

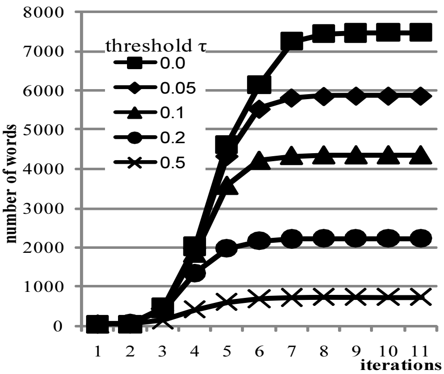


Figure 2. Change in the number of words registered in the dictionary with successive iterations

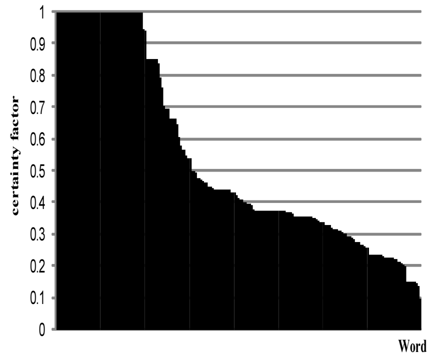


Figure3. Distribution of the certainty factor of the extracted words

Figure.2 shows the sum of the number of words in the feature expression dictionary and the number of words in the evaluation expression dictionary, indicating how they increase with successive iterations, for cases where the pre-set threshold value, τ, is 0, 0.05, 0.1, 0.2, and 0.5. As the pre-set threshold value, τ, increases, so the number of words registered in the dictionaries decreases. In each case, at the start of updating, the number of words in the dictionaries increases. However, by the eighth iteration the number of words has stabilized. Figure.3 shows the certainty factors f(Wk) and e(Wk) of the words registered in the third iteration when the pre-set threshold value, τ, is zero. The X axis of Figure.3 shows each word ranked in descending order of certainty factor. 25% of the words have a certainty factor of 1.0, but the certainty factor of the remaining words decreases rapidly to a level less than 0.4 and then slowly to 0.1. High credibility dictionaries can be built by registering only words which have a certainty factor of 1.0. However since in this case the 75% of words represent the reputation information extracted by three types of the syntax structure pattern are discarded and used to extract reputation information, much valuable information may not be obtained.

Figure 4 shows the relation between the precision ratio of the evaluation expression and the iteration in the generation of the evaluation dictionary. The

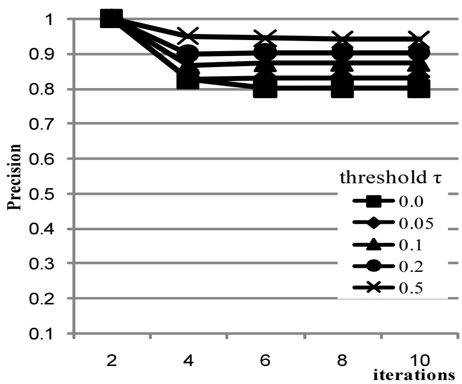


Figure.4 Precision ratio of the evaluation expression

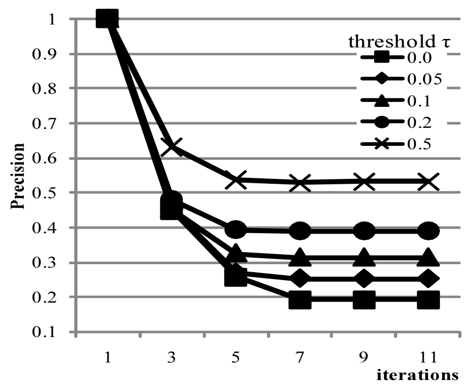


Figure.5 Precision ratio of the feature expression

precision ratio obtained by our proposed method which registers only the words which have a certainty factor greater than the pre-set threshold value, τ, is higher than obtained by the method of registering all extracted words.

Figure 5 shows the relation between the precision ratio of the feature expression and the iteration in the generation of the feature dictionary. Even though the precision ratio declines rapidly as the number of iterations in the generation of the feature dictionary increases, the precision ratio obtained by our method is still higher than obtained by the method of registering all extracted words.

It seems not to be realistic to obtain the recall ratio on the information retrieval from a vast number of Web documents. However, in order to evaluate the completeness of the information retrieval system, in addition to the precision ratio, the recall ratio should be considered. We substitute recall with the number of the words registered in the dictionary, so a maximum number of the appropriate expression that there is in a dictionary is recall ratio 100%. Then the F-measure, which represents the completeness of the information retrieval system, is obtained as shown in equation 1:

$$F - measure = (2 \times precision \times recall) \div (precision + recall) \quad (1)$$

Figure 7 shows the relation between the F-measure of the feature expression dictionary and the iteration number. The dictionary built with at least 5 iterations, using a pre-set threshold value, τ, of 0.05, 0.1, or 0.2, achieved a higher F-measure than the dictionary built with no pre-set threshold value. Consequently it can be concluded that our proposed method is useful in achieving a high F-measure. However in case of the pre-set threshold value, τ, of 0.5, the F-measure is lower than for the other cases, due to the large number of discarded words. Therefore there may be an optimal threshold value, τ.

4. Conclusions

A method of building a highly credible dictionary has been established. The principle of our method is based on obtaining the confidence factor of three types of syntax structure patterns, assigning words a certainty factor, as a measure of how certain it is that the words really do describe an evaluation expression or feature expression, and

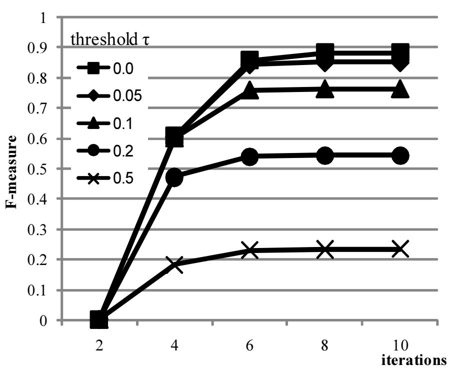


Figure.6 F-measure of the evaluation expression

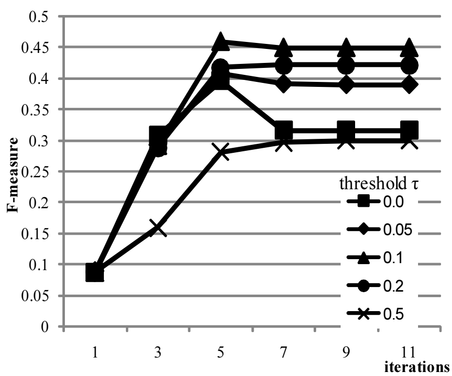


Figure.7 F-measure of the feature expression

registering only words that have a certainty factor greater than a pre-set threshold value. However the credibility of the feature dictionary is not necessarily high enough. Future studies to increase the credibility of the feature dictionary and obtain highly reliable reputation information are described as follows:

(1) Specialized information filtering for an object in an analysis might be able to increase the precision ratio of the feature expression dictionary.

(2) 33 syntax structure patterns have been defined by the combination of the three construction patterns and different parts of speech. The reason for rapid saturation of the number of the words registered in the dictionaries in the early stages of updating is the lack of completeness of the syntax structure patterns. In order to increase the recall ratio of the proposed method, more construction patterns are required.

(3) Before extracting the reputation information from the Web, the correction of sentences which have grammatical faults is required since the documents obtained from the Web do not abide by Japanese language grammar rules.

Acknowledgements

This research is supported by "Tangible Software Engineering (SE) Education and Research." as part of the "Program promoting the levelling of private university academic research," in which the Ministry of Education, Culture, Sports, Science and Technology invited public participation for the academic year 2007.

References

[1] "The findings of Internet shopping intention from 10,000 people"[online], Available from: research.goo.ne.jp/ [Accessed 15/04/2008].(in Japanese)

[2] Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, Toshikazu Fukushima: Collecting Evaluative Expressions for Opinion Extraction, In Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04), pp. 584-589, 2004

[3] Shigeru FUJIMURA, Masashi TOYODA and Masaru KITSUREGAWA, "Extracting Evaluative Expressions and Reputations from the BBS", JSAI2004, 3F1-03, 2004.(in Japanese)

[4] Taichi Nakamura and Hiroshi Maruyama: Extracting Opinions Relating to Consumer Electronic Goods from Web Pages",JOINT CONFERENCE ON KNOWLEDGE-BASED SOFTWARE ENGINEERING 2006(JCKBSE'06), IOS Press, 2006, pp.206-209, 2006

[5] Peter D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews", In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 417-424, 2002.

[6] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques" Empirical Methods in Natural Language Processing (EMNLP2002), pp.76-86, 2002.

[7] Kushal Dave, Steve Lawrence, and Dvid M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews", In Proceedings of the 12th International World Wide Web Conference, pp. 519-528, 2003.

[8] Takashi Inui, Manabu Okumura: A Survey of Sentiment Analysis, Journal of Natural Language Processing, Vol.13, No.3, 2006.(in Japanese)

[9] Kenji TATEISHI, Yoshihide ISHIGURO, Toshikazu FUKUSHIMA, "Opinion Information Retrieval from the Internet", In IPSJ SIGNL. Note 144-11, pages 75-82, 2001. (in Japanese).

[10] "Evaluative expressions"[online], Available from: http://www.syncha.org/evaluative_expressions.html [Accessed 15/04/2008](in Japanese)

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-359

Mining Combinatorial Effects on Quantitative Traits from Protein Expression Data

Takuya YOSHIHIRO*†, Etsuko INOUE† and Masaru NAKAGAWA†
†*Faculty of Systems Engineering, Wakayama University*

Abstract. In this paper we proposed a data mining method to find interesting relation among the small number of proteins and a quantitative trait from protein expression data. It is practically important to find small number of proteins which in combination effects on some characteristics of samples because it motivates further experiments to investigate the function of the proteins. Further, treating quantitative trait rather than categorical data is also practically important since considerable part of characteristics of samples is represented by quantitative values. Our data mining method apply discriminant analysis into protein expression data and quantitative trait data to derive interesting relations among them.

Keywords. Data Mining, Bioinformatics, Proteins, Expression

1. Introduction

Since the whole DNA sequences have been in public, many post-genome researches begin to investigate the system of living creatures. Creatures consist of vast sort of proteins and their bodies and are maintained by their complex interaction of genes and proteins. One of the major interests in this area is that how the characteristics of each individual appear and what kind of proteins are related with it.

One major approach to clarify the mechanism of living creatures is to derive some patterns or rules from expression data which imply some interactions of genes or proteins. In fact, much work from this approach has been done with gene expressions of microarrays [1]. Microarrays are so useful because we can treat expressions of thousands of genes simultaneously. Several statistical and informational techniques are applied for expression data to find interactions among genes and further to infer their interaction networks. Bayesian networks [2] and Boolean networks [3] are the typical strategy among them and they have been contributed to predict functions of genes.

* Corresponding Author
† Faculty of Systems Engineering, Wakayama University, 930 Sakaedani Wakayama, 640-8510, Japan. E-mail: {tac, etsuko, nakagawa}@sys.wakayama-u.ac.jp.

Applying machine learning methods into expression data is also one of major approaches to predict functionality of genes and proteins. Some work tries to classify genes with its functionalities by finding groups of genes which expression patterns are similar. Since genes or proteins of similar functions tend to be expressed similarly, those approaches contribute to predict their functions. Typically, those machine learning based approaches involve separation of multi-dimensional space into several regions so that we can obtain some knowledge about the region that samples of each class belong to. In practice, however, this approach derives too rough information to motivate deeper experimental analysis because it usually shows the information of so many dimensions. To find the smaller number of genes or proteins which effect on a sample character is important to clarify the cellular networks of living creatures since it enables experimental researchers to concentrate on the particular genes or proteins to investigate deeply. In addition, from the practical point of view, it is also desired to treat quantitative traits rather than categorical traits as characteristics of samples because many characteristics are in many cases represented as quantitative values.

In this paper, from consideration of the practical requirements shown above, we try to find combinations of small number of proteins which have relation with a quantitative trait based on linear discrimination. Although SVMs are frequently used for linear (and non-linear) separation of samples of microarray data [4] (and also a trial to find small number of genes exists [5]), it is not suitable for protein expression data because protein expression data tend to include a few burst noise values coming from the experimental process of obtaining data (as shown in Section 2).

Thus we propose a data mining method for such data and develop a tool based on it using statistical discriminant analysis to extract the (small) combinations of proteins which effects on quantitative traits. With our mining tool, we succeeded to found several interesting combinations of proteins from real protein expression data which imply the possibility of protein interactions.

The rest of the paper is organized as follows. In Section 2, as a preparation, we describe the operation of obtaining protein expression data and also give explanation about discriminant methods in the literary. In Section 3, we describe the mining method proposed and its implementation as a practical mining tool. In Section 4, we evaluate our method by applying real protein expression data and quantitative trait data. We give some discussion in Section 5 and finally in Section 6 we conclude the work.

2. Linear Discrimination in Protein Expression Data

2.1. Protein Expression Data and Quantitative Traits

In this section, we explain the operational process to obtain protein expression data and also mention that the data sometimes include burst noise value compared with gene expression data of microarrays.

Protein expression data represent the amount of each protein i of sample j. Typically, the number of proteins in the data are about several hundreds to thousands and the number of samples is at most hundreds. The process of obtaining protein expression data in this work is somehow complicated compared with microarray data (Fig. 1). First, we obtain 2-dimensional electrophoresis image from each target sample through biological experimental processes. Second, we identify spots of separated proteins using image processing software and compute the expression level of each spot. Third,

we match the spots of the same protein among the images in the experiment. Finally, we normalize the values of expression levels using one of the normalization methods to throw it into data mining processes.

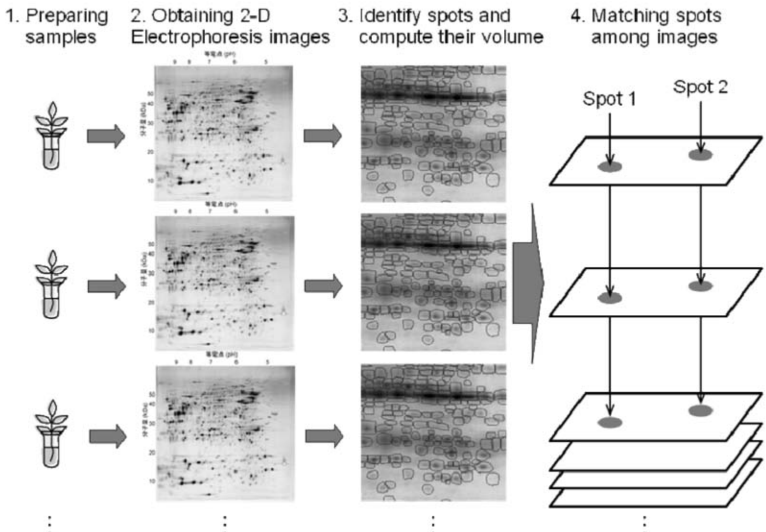


Figure 1. The process of obtaining proteome expression data.

In the process of obtaining protein expression above, there are larger number of factors which cause noise in expression values compared with gene expression of microarrays. Note that in the case of protein we have to separate proteins into individual spots while the gene spots in microarrays are originally independent. Therefore, in case of protein expression, several spots sometimes join into an identical spot and also a spot may include different proteins. In practice, although we always do the operations so careful to obtain precise protein expression values, noise in the protein data always becomes larger than gene expression of microarrays. Several spots sometimes even overlap to be mixed into an identical spots together. As a result, we find some expression level far larger or far smaller than it is expected.

In this paper, we handle two kinds of data: one is protein expression data as described above and another is quantitative trait data. Fig.2 shows an example of those data. The protein expression data are shown in Fig. 2(a), the form of which is an ordinary 2-dimensional table and it represents expression levels of each protein at each sample. Fig 2(b) shows the quantitative trait data. The data is also shown as an ordinary 2-dimensional table and each trait value represents some characteristics of samples.

	protein A	protein B	protein C	…
sample 1	1.534	3.972	4.081	
sample 2	1.587	2.651	3.761	
sample 3	1.793	3.552	3.623	…
⋮			⋮	

(a) protein expression

	trait a	trait b	trait c	…
sample 1	23.4	3	4.231	
sample 2	21.4	6	2.123	
sample 3	27.9	4	3.623	…
⋮			⋮	

(b) quantitative traits

Figure 2: The data forms for data mining

2.2. Linear Discriminant Methods

We try linear discrimination of two classes of samples in 2-dimensional space of two proteins. One of the major methods for discrimination is called SVM (Support Vector Machine) which is often used in machine learning applications. SVM computes a linear function to separate two classes of samples efficiently. Fig. 3 shows the simple example of SVM. In Fig. 3(a), the linear function completely separates the two groups in the 2-dimensional space. When linear separation is impossible, SVM introduces soft margin, within which samples of both classes are allowed to be mixed (Fig. 3(b)). SVM computes the linear function which minimizes the soft margin. However, in case of presence of burst noise shown in Fig. 3(c), namely the case that there is a sample which has exceptional noise expression level, SVM cannot find the function to separate the two classes clearly. Since protein expression data tend to include this kind of noise samples, we try to find linear separation function using statistical discriminant analysis.

Discriminant analysis first computes the center of the two classes and next computes the linear function which separates the two centers around the middle of them. Fig. 3(d) shows that discriminant analysis finds the linear function to separate in the same case as Fig. 3(c). As shown here, statistical discriminant analysis is suitable to use in this case.

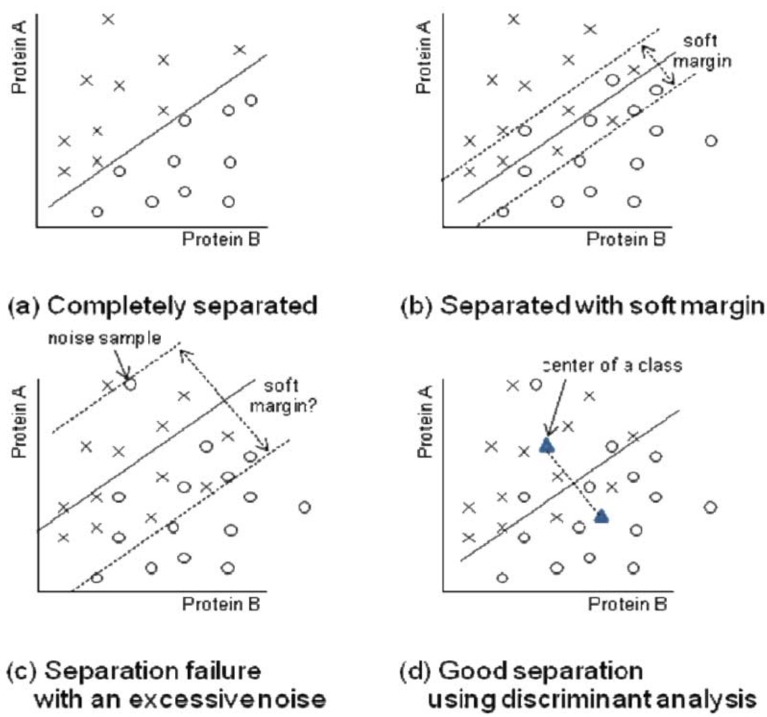


Figure 3. Linear Discrimination Methods

3. Data Mining Method and Systems

3.1. Method

Our data mining method consists of three folds: (1) compute separation score of discriminant analysis for all combinations of proteins and thresholds, (2) make a ranking with the separation score to pick up well-separated results, and (3) display scatter diagram to judge whether the results are interesting or not (Fig.4). In our mining process, the human confirmation phase (step (3)) is included because we consider that interesting results are finally determined by not only a simple value such as separation score but by a human confirmation. For this reason we developed a practical tool to support such analytical operations.

In step (1), we pick two proteins up from the expression data and decide a threshold value for a quantitative trait. The threshold value is selected from several candidates of threshold values for each trait and we classify samples into two classes with this threshold value. We try every combination of two proteins and a threshold value, and then for each combination we compute its separation score which represents the clearness of the separation result. Concretely, the score is the ratio of samples classified correctly by the discriminant function. So, if the two classes are completely separated by the linear function, the separation score is 100%. High separation score means that the separation is well-done so that the result is interesting with high probability.

In step (2) of our mining process, we create a ranking list of the combinations ordered by the separation scores. This ranking is made for each quantitative trait. Then in step (3), users check the scatter diagrams one by one from higher rank to pick up interesting combination of proteins.

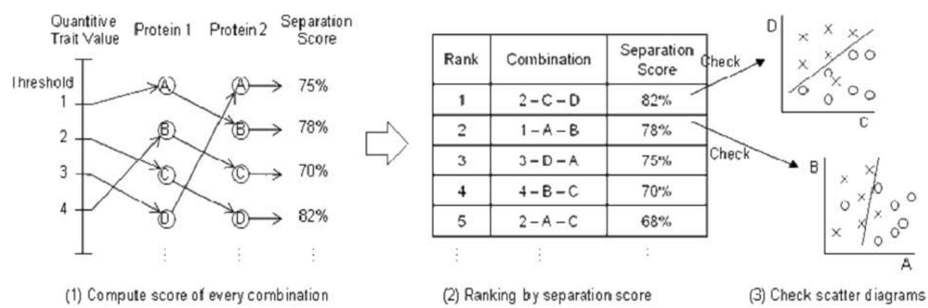


Fig. 4 Data Mining Operations

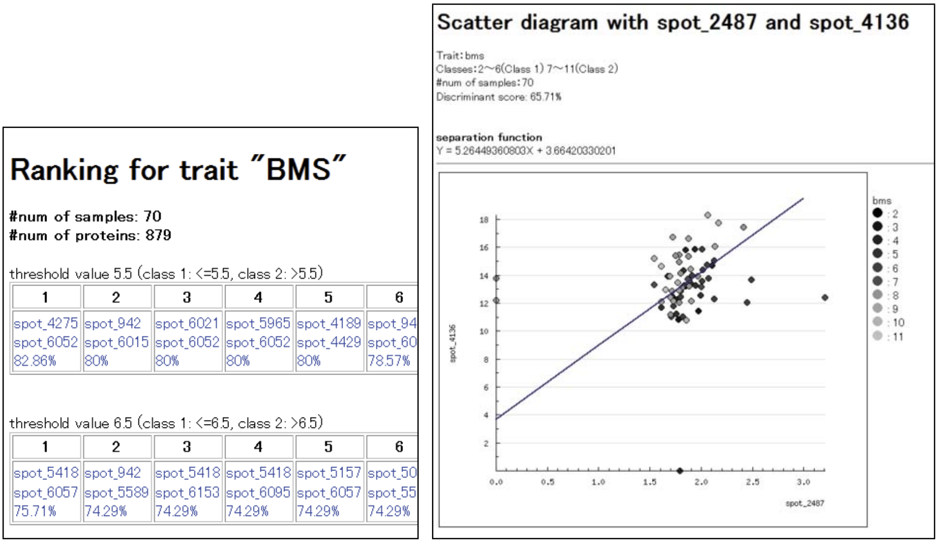
3.2. Implementation

We developed a software system to be used as a mining tool of our method. Our system works in two phases; the first is computation phase with PCs while the second is viewing phase by users. In its computation phase, the system computes separation scores for each combination of step (1) (of the mining operation shown in Fig. 4), and save the results into files. Then, in the second phase, we create ranking so that users can check each scatter diagram one by one whether the diagram shows interesting

relation among two proteins and a quantitative trait. Namely, step (1) of our method corresponds with the computation phase and step (2) and (3) do with the viewing phase.

The system is developed as a web application using Linux, Apache and PHP[6]. In addition, statistic software R [7] is also used in its computation phase.

Fig. 5(a) shows the interface where the ranking for each threshold values on a quantitative trait. If we click each column of the ranking, its scatter diagram appears to check the distribution is interesting or not (Fig. 5(b)). In this scatter diagram, quantitative trait values are shown with gradation colors so that users can easily grasp the whole trend of separation.



(a) Ranking with separation scores (b) Displaying scatter diagram

Fig.5 Examples of user interface of the system

4. Evaluation

We evaluate the proposed method and the implemented system using real protein expression data. The data includes 70 samples, 10 quantitative traits and 879 proteins of Japanese Beef cattle. As quantitative traits, we use several values on beef quality. Note that several quantitative traits are real numbers but some of them takes several integers in a constant range (e.g., 1, 2, …, 12).

As for computation time, it takes about 3 hours to test all combinations of protein pairs for each threshold value on a quantitative trait. Since we try 6 threshold values for each trait, it takes about 18 hours for each trait.

As a mining result, we found several interesting combinations of two proteins where the scatter diagram shows clear separation trend. Fig. 6 shows the example of such diagrams where the separation is clearly shown although a few burst noise is seen. Such results are to be noted to biological experimental researchers to do further investigation of finding mechanisms of the relation.

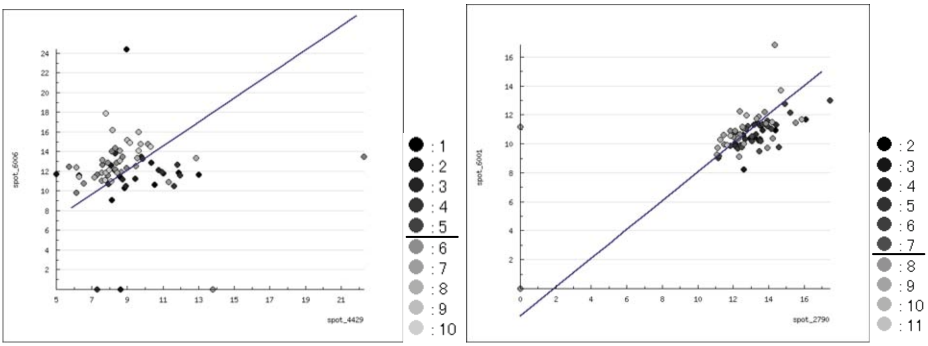
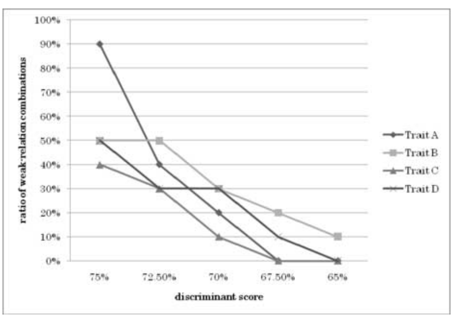


Fig. 6. Examples of clearly separated pair of proteins.

From the result, we further found several combinations which cannot clearly be said to be separated, but they seem to include weak trend of separation. Then, we count the number of such weak-related combinations for each range of separation scores, where we judge the weak relation combination subjectively. We discuss later about important points in this judgment. Fig. 7(a) shows the results where the ratio of weak-related combination for each score from 60% to 80%. We found that most of the weak-related combinations have the score of more than 70%. We did the same for four traits of beef quality but the whole trend is found to be almost the same. On the other hand, Fig. 7(b) shows the distribution of the separation score of different threshold values for a trait which takes the integer values between 1 and 12. We find that the score goes high when the threshold value comes far from center value 6.5. It implies that even if we want to check almost all weak-related combinations, we do not have to check too many combinations, i.e., we have to check about hundreds of them.



Trait A

threshold value 5.5

score	～80%	80%～75%	75%～70%	70%～65%	65%～60%	60%～	total
#num	5	274	4032	32331	61520	287719	385881
ratio	0.00130%	0.07101%	1.045%	8.378%	15.943%	74.562%	100%

threshold value 6.5

score	～80%	80%～75%	75%～70%	70%～65%	65%～60%	60%～	total
#num	0	1	316	11430	52674	321460	385881
ratio	0%	0.00026%	0.082%	2.962%	13.650%	83.305%	100%

threshold value 7.5

score	～80%	80%～75%	75%～70%	70%～65%	65%～60%	60%～	total
#num	0	2	184	8688	42246	334761	385881
ratio	0%	0.00052%	0.048%	2.251%	10.948%	86.752%	100%

threshold value 8.5

score	～80%	80%～75%	75%～70%	70%～65%	65%～60%	60%～	total
#num	1	325	4757	35986	70603	274209	385881
ratio	0.00026%	0.08422%	1.233%	9.326%	18.297%	71.061%	100%

(a) the ratio of weak-related combinations (b) score distribution for a trait

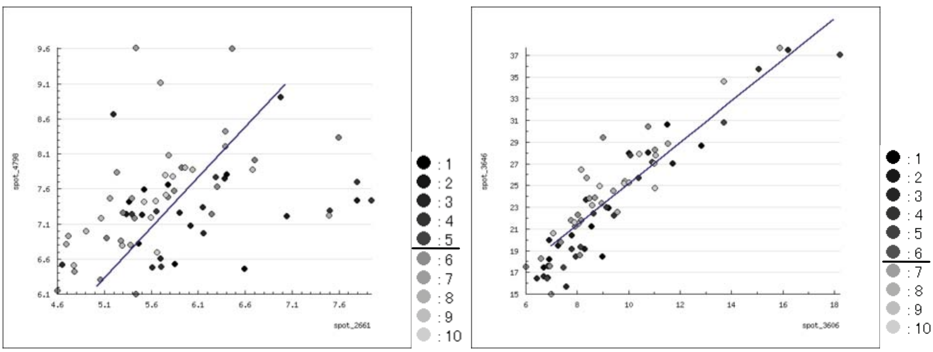
Fig. 7. Distribution of the separation scores and weak-related combinations

5. Discussion

In this work we try to pick up interesting relation between a quantitative trait and a pair of proteins from protein expression data using discriminant analysis. We support several practical requirements, i.e., treating quantitative traits, deriving combinations of small number of proteins, and treating expression data with a few burst noise values.

As a result of our trial, we obtain several clearly separated combinations which infer some relation among them. Also we present an overview of the distribution of weak-related combinations.

Weak-related combinations are difficult to utilize practically because it gives us a little but not enough relation to motivate further study on them. However, it is certain that they may include valuable knowledge about functions of proteins. From our trial, the possibility is implied that we can find interesting combinations even if the separation score is not particularly high. In fact, some interesting combinations take not so high score while considerable numbers of high-score combinations are not so interesting. See Fig. 8 where we see two scatter diagrams of the same score. Here, (a) is an example of the combinations that we judge it includes weak relation, but (b) does not include any separation trend. It is considered difficult to find the conditions that determine the interest factor of diagrams. It is future work to find the conditions and raise the rank of interesting combinations.



(a) A weak-related combination (b) A non-related combination

Fig. 8. A weak-related and a non-related combination of the same score (77.14%)

6. Conclusion

In this paper, we proposed a data mining method to find well separated combinations of proteins with respect to quantitative traits. We support several practical requirements of treating quantitative traits, deriving combinations of small number of proteins, and treating expression data with a few excessive noise values. We also implemented them and evaluated with real protein expression data. As a result, we found several clearly separated combinations of proteins and further discussed about the distribution trend of interesting combinations. This work tried to develop a practical tool which is useful for experimental researchers. We think that a part of this objective is successfully achieved, but further work to improve the accuracy of finding interesting diagrams is desired.

Acknowledgment

We really appreciate Keiju Yokoyama for progressing analytical work devotedly. This study was partially supported by a grant from the Wakayama Prefecture Collaboration of Regional Entities for the Advancement of Technological Excellence of JST (Japan Science and Technology Agency).

References

[1] J. L. DeRisi, V. R. Lyer, P. O. Brown, “Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale,” Science, 278, 680—686, 1997.

[2] S. Imoto, T. Goto and S. Miyano, “Estimation of genetic networks and functional structures between genes by using Bayesian network and nonparametric regression, Pacific Symposium on Biocomputing, 7, 175-186, 2002.

[3] T. Akutsu, S. Kuhara, O. Maruyama and S. Miyano, Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model *Theoretical Computer Science* **298**, 235-251, 2003.

[4] L. Wang, F. Chu and W. Xie, “Accurate Cancer Classification Using Expressions of Very Few Genes,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 4, No. 1, pp.40—53, 2007.

[5] Y. Lu, F. Liu, M. Sanchez and Y. Wang, “Interactive Semisupervised Learning for Microarray Analysis,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol.4, No.2, pp.190—203, 2007.

[6] PHP, http://www.php.net/.

[7] R, http://www.r-project.org/.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-368

Extraction of Children Friendship Relations from Activity Level

Shigeo KANEDA [a,1], Takuya KATSUKI [a], Masahiro SHIBATA [a], Hirohide HAGA [a], Mari UEDA [b], Aki KONO [c], and Kimio SHINTANI [c]

[a] *Faculty of Engineering, Doshisha University*
[b] *Doshisha Girl's Junior and Senior High School*
[c] *Department of Early Childhood Education, Tokiwakai College*

Abstract. Children learn to fit into society by group interaction through which they are greatly influenced by their relations with friends. Although teachers need to observe them to assist their growth and social progress and support the development of their personalities, only experienced teachers can watch children while simultaneously providing high-quality guidance. To resolve this problem, this paper proposes a mathematical and objective method that assists teachers with observation. It uses the numerical data of activity levels recorded by pedometers from which we make tree diagrams called dendrograms based on hierarchical clustering with recorded activity levels. We also calculate the children's "breadth" and "depth" of friend relations using more than one dendrogram. After the authors recorded the children's activity levels at a kindergarten for two months to evaluate our proposed method, we found that the results generally coincided with teacher remarks and evaluations of the children.

Keywords. Activity level, Association, Clustering, Data minig, Lifelog

Introduction

In nursery schools and kindergartens, children learn to fit into society by group interaction through which they are greatly influenced by their relations with friends. Although preschool teachers need to observe them to assist in their growth and social progress and support the development of their personalities, only experienced teachers can watch over children while providing high-quality guidance.

Another issue is becoming a burden for young and inexperienced teachers, *children with special needs* who cannot make friends even though their development level of intelligence is average[2]. Teachers have to identify such special needs children in their early

[1]Corresponding Author: Doshisha University, 1-3, Miyakodani, Tatara, Kyotanabe-city, 610-0321, Japan; E-mail: skaneda@mail.doshisha.ac.jp

[2]Children with special needs include: 1) unique character or illegal condition at his/her home; 2) Autism-Asperger syndrome; 3) LD-Learning Disability; or 4) ADHD-Attention Deficit Hyperactivity Disorder. A Japanese Government survey in 2002 claimed that 6.3% of elementary and junior high students are special needs.

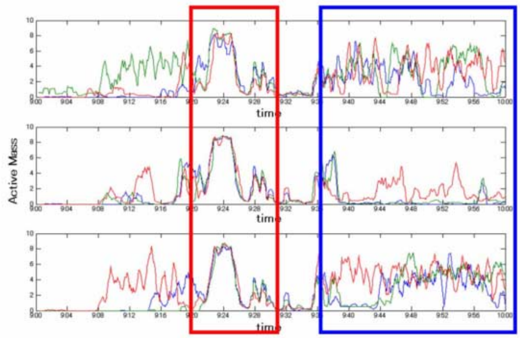


Figure 1. Activities of nine children from 9:00 to 10:00 a.m.

stages of development to properly support their development. However, this responsibility is very hard for young and inexperienced teachers.

In a previous study[1][2], the authors demonstrated a new approach to analyze friend relations by using activity levels, recorded by pedometers, and tree diagrams called dendrograms based on hierarchical clustering. Pedometers detect the children's activity every two minutes. For instance, assume that children are playing on the playground for 20 minutes. The pedometer data will have just 10 points, which are too few for clustering calculations. Thus, the proposed method is only applicable to observations that last more than one day. Two-minute intervals are not adequate for detailed analysis.

Thus, this paper proposes a new approach to obtain more accurate and detailed behavior of the children using new pedometers that detect their activity every four seconds. The proposed method finally clarifies the "breadth" and "depth" of friend relations.

1. Proposed Method

The authors used new Lifecorder EX pedometers manufactured by Suzuken Corp. to detect activity levels every four seconds. Since this pedometer increases the data density by 30-fold, the authors could extract pure free play interval data from the all-day data for friend relation analysis. Also, a new approach is proposed to determine the "depth" and "breadth" of child relations using the data of several days. The following shows the outline of the proposed method.

Step 1: Activity Measurement Each child wears a pedometer. The measured data are copied into a PC.

Step 2: Clock Adjustment The times of all pedometers are aligned.

Step 3: Data Extraction Free play interval data are extracted.

Step 4: Clustering Process Friend relations are calculated from the extracted data by a clustering algorithm.

Step 5: Depth and Breadth These are obtained from the multiple day analysis and are defined below.

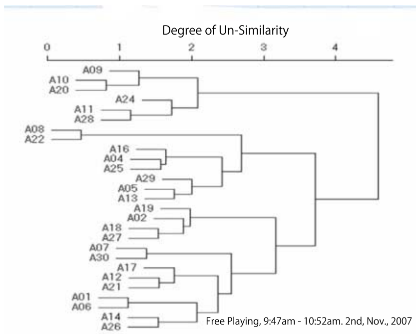


Figure 2. A dendrogram in free play (five year olds, Ward method).

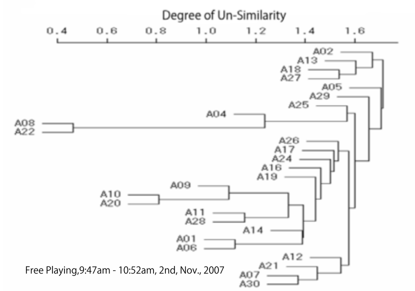


Figure 3. A dendrogram in free play (five year olds, nearest neighbour method).

2. Evaluation in Actual Kindergarten

To verify the effectiveness of the proposed method, the authors made the social experiment. Observation records were written everyday about students, because the kindergarten schedule often changes unexpectedly.

Figure 2 shows a clustering result using the Ward method and the activity records of Nov. 2. Fig. 3 also shows the result using the nearest neighbor method for the same data. Statistical package R[7] is employed for the calculation. The Ward method, which minimizes the increase of the sum of squares in each group, is robust and suitable to make groups. On the other hand, the nearest neighbor method usually generates long chain-like clusters. To evaluate the influence of the clustering algorithm, we employed these two typical clustering algorithms.

In Figs. 2 and 3, the movement of children A08 and A22 are very similar. Their teacher describes A08 as a potential special needs child. A22 is very interested in A08 and chases after A08. However, the authors in an engineering domain could not visually recognize that A08 might be a special needs child. Only the preschool home room teacher correctly recognized A08's special behavior. The dendrogram and the preschool teacher's observation correspond, suggesting that our proposed method may become a support tool for inexperienced teachers to grasp friend relations.

A difference can be found between the results of Figs. 2 and 3. In Fig. 2, there are many similar sized clusters. This is a typical result of the Ward method because it is suitable for grouping. On the other hand, there is only one big cluster in the nearest neighbor method dendrogram, as shown in Fig. 5. The nearest neighbor method clarifies

who is farthest from the other children: that is, playing alone. The method may detect a child with special needs.

The clustering result provides good information for observing friend relations in a class. However, the results possess lower reliability if only one-day data are analyzed. Thus, in this subsection, we introduce two new concepts: *depth and breadth*. Depth evaluates how deep one child's companionship is with other children. Breadth represents how wide one child's companionship is with the other children. The following steps are proposed.

Step 1: Measurement and Dendrogram Generation

Children wear each pedometer as long as one month, and dendrograms are generated for each free play interval.

Step 2: Grouping

Each dendrogram is divided into eight *sub-clusters*, based on Parten's theory that argues that children make groups of three or four members. The total number of children (29) is divided by three or four. Eight is basically the result of this calculation.

Step 3: Counting

Let K_i be an child, where $i = 0, 1, 2, \cdots, n$, and n are the number of children in the class. Let $p(K_i, K_j)$ be a combination of children K_i and K_j, where $i \neq j$[3]. For each i, ($i = 1, 2, \cdots, , n$), make all $p(K_i, K_j)$, where $j = 1, 2, \cdots., i-1, i+1, \cdots., n$. Count the number of $Count(i, j)$ for each j, where $Count(i, j)$ is the frequency that children K_i, and K_j are in the same sub-cluster in each divided dendrogram for all free play intervals. For instance, assume 28 free play intervals. If children K6 and K7 appear in the same sub-cluster in 12 of 28 free play intervals, the $Count(6, 7) = 12$.

Step 4: Calculation of Depth and Breadth

Depth and breadth are calculated as follows.

Breadth: The breadth of child K_i is the median of all $Count(i, j)$ for $j = 1, 2, \cdots, i-1, i+1, \cdots.., n$. If the median is large, the child is always in large groups. On the other hand, if the median is small, then the child spends much of his/her preschool life in small groups. For instance, if $n = 8$ and $Count(6, 1) = 3$, $Count(6, 2) = 3$, $Count(6, 3) = 4$, $Count(6, 4) = 4$, $Count(6, 5) = 4$, $Count(6, 7) = 5$, $Count(6, 8) = 5$, then the breadth is four.

Depth: The depth of child K_i is a maximum value in all $Count(i, j)$ for $j = 1, 2, \cdots, i-1, i+1, \cdots.., n$. If this maximum value is large, this child always spends time with a specific child or children. These children always play together in a specific group. For instance, if $n = 8$ and $Count(6, 1) = 1$, $Count(6, 2) = 1$, $Count(6, 3) = 2$, $Count(6, 4) = 2$, $Count(6, 5) = 3$, $Count(6, 7) = 3$, $Count(6, 8) = 6$, then the depth is six.

[3]For instance, $p(A08, A22)$ is a pair for children, $A08$ and $A22$.

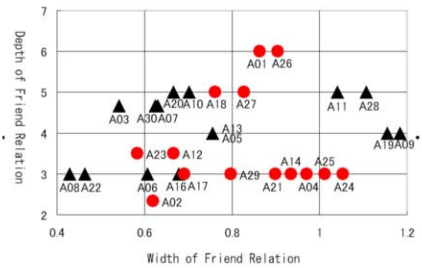


Figure 4. Breadth and depth results (five year olds, Ward method).

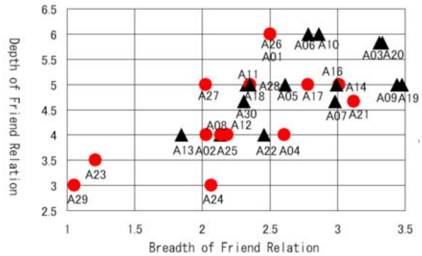


Figure 5. Breadth and depth results (five year olds, nearest neighbour method).

3. Conclusion

This paper proposed a new tool to grasp the friend relations of children in kindergartens or nursery schools using pedometers and hierarchical clustering. The authors performed a social experiment at a kindergarten in Osaka for four weeks in 2007 in a class of 29 five-year olds. The obtained dendrogram agreed with the visual observations of the experienced teacher.

This paper also proposed a new calculation method of the depth and breadth of children friend relations using multiple dendrograms. These results determine the tendency and development of each child in his/her preschool life. This result also showed good agreement with the observations of an expert teacher. Finally, the proposed methods are applicable to support young and inexperienced preschool teachers.

References

[1] Kono, A., Shintani, K., Ueda, M., and Kaneda, S., "New Childhood Observation method by Active Mass," in Proc. of the the 4th IASTED International Conference on Web-based Education, No.461, 2005.

[2] Kumon, S., Kohno, A., Shintani, K., Kaneda, S., and Haga, H. " Research for Analysis of Companion Relation Based on Measurement of Action," in Proc. of PECERA (Pacific Early Childhood Education Research Association) 7th Conference, pp.168–169, 2006.

[3] Hyun, E. "Making sense of developmentally and culturally appropriate practice (DCAP) in early childhood education," Peter Lang., Chapter 2, New York, 1998.

[4] http://www.suzuken.co.jp/english/index.html

[5] http://www.suzuken.co.jp/english/whats/closeup/diabetes01.html

[6] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data," Prentice Hall, 1988.

[7] The R Project for Statistical Computing, http://www.r-project.org/

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-373

Mining of Protein Subcellular Localizations based on a Syntactic Dependency Tree and WordNet

Mi-Young Kim
School of Computer Science and Engineering, Sungshin Women's University, Dongseon-dong, Seongbuk-gu Seoul, 136-742, South Korea
miykim@sungshin.ac.kr

Abstract. Detection of protein subcellular localization is essential in information extraction from biomolecular texts. There has been a great deal of research on text mining to detect protein subcellular localization information in documents. Previous researches have insisted that linguistic information is useful for identifying the subcellular localizations of proteins. However, previous systems for detecting protein subcellular localizations have used only shallow syntactic parsers, and showed poor recall. Thus, there remains a need to use a deep level of linguistic knowledge to the analysis of text. To improve performance in detecting protein subcellular localization information, this paper proposes a method based on a syntactic dependency tree and WordNet. From the syntactic dependency tree, we construct syntactic paths from a protein to its location candidate. Then, we retrieve syntactic and semantic information from the root, protein subtree and location subtree of each syntactic path. We extract syntactic category and syntactic direction as syntactic information, and synset offset of the WordNet thesaurus as semantic information. According to the information, we extract (protein, localization) pairs. Even with no biomolecular knowledge, our method shows reasonable performance in experiments using Medline abstract data. The experimental results show that our method outperforms previous methods, and the obtained syntactic and semantic information contributes to the improvement of the performance.

Keywords. protein subcellular localization, syntactic dependency tree, WordNet, data mining, bioinformatics

1. Introduction

Information extraction from the biomedical literature has become an active area of research. Rather than using machine learning methods based on simple features, many researchers now use linguistic information in analyzing large amounts of biomedical data. Various studies have insisted that such linguistic information is useful for information extraction from biomedical data. To detect protein interactions, some previous studies applied syntactic information using a full syntactic parser, and showed improved performance[1,2]. However, to detect protein subcellular localization information, previous studies have used shallow partial parsers. Such partial parsers

detect only phrase boundaries, and do not provide full syntactic information about biomedical text. It is necessary to retrieve deep linguistic features for detection of protein subcellular localization information.

To improve performance in detecting protein subcellular localizations, we propose a method based on a syntactic dependency tree and WordNet thesaurus. We obtain full syntactic dependency trees to Medline abstract data. In addition, we attach the synset offsets for the nodes of the syntactic dependency trees as semantic information. Our experiments show that the proposed method significantly outperforms existing methods. We also describe the contribution of the information used in its performance.

2. Previous Work

The task of relation mining in the biomedical domain has been studied extensively in recent years[1, 3]. In detecting protein subcellular localization, many previous researches have focused on machine learning methods, based on protein sequence data. As the volume of biomedical text has increased, some researches have attempted to apply text mining simply using word features[3, 4].

To improve performance, recent studies used linguistic information as a key to improve performance. Craven and Kumlien[5] performed linguistic process using the Sundance partial parser. They applied a syntactic partial parser that detected only phrase boundaries. Skounakis *et al.*[6], Page and Craven[7] applied a hierarchical hidden Markov model (HMM) method using context features. They constructed a model using the same Sundance shallow parser. Goadrich *et al.*[8] also applied the Sundance partial parser and obtained information on 251 predicates from the parsing results, which usually indicated phrase position information. They reported precision of 0.58, recall of 0.40, and F-measure of 0.47, which was the best result for the experimental data reported by Craven and Kumlien[5].However, the recall is still low.

We built on the conclusion of the previous work that linguistic information, especially syntactic information, is an important key for detecting protein subcellular localization information. However, previous studies applied only shallow parsers and simply detected phrase boundaries. A deeper level of linguistic information is necessary. Thus, based on a full syntactic dependency tree and WordNet thesaurus, this paper proposes the detection of protein subcellular localization, and we report improved performance using syntactic and semantic information.

3. Mining of Protein Subcellular Localization based on a Syntactic Dependency Tree and WordNet

Syntactic and semantic information is used as linguistic cues. For syntactic information, we apply the MINIPAR syntactic dependency parser[9]. Using the MINIPAR parsing result, we can draw a syntactic dependency tree as shown in Figure 1. For syntactic information, syntactic category and syntactic direction information is extracted from the MINIPAR results. Syntactic direction information is one of two values, gov and dep, indicating whether the current node is the governor or the dependent of the previous node, respectively. As semantic information, the synset offset of WordNet is employed. Synset offset is a number that uniquely identifies a concept node(synset) in the WordNet hierarchy.

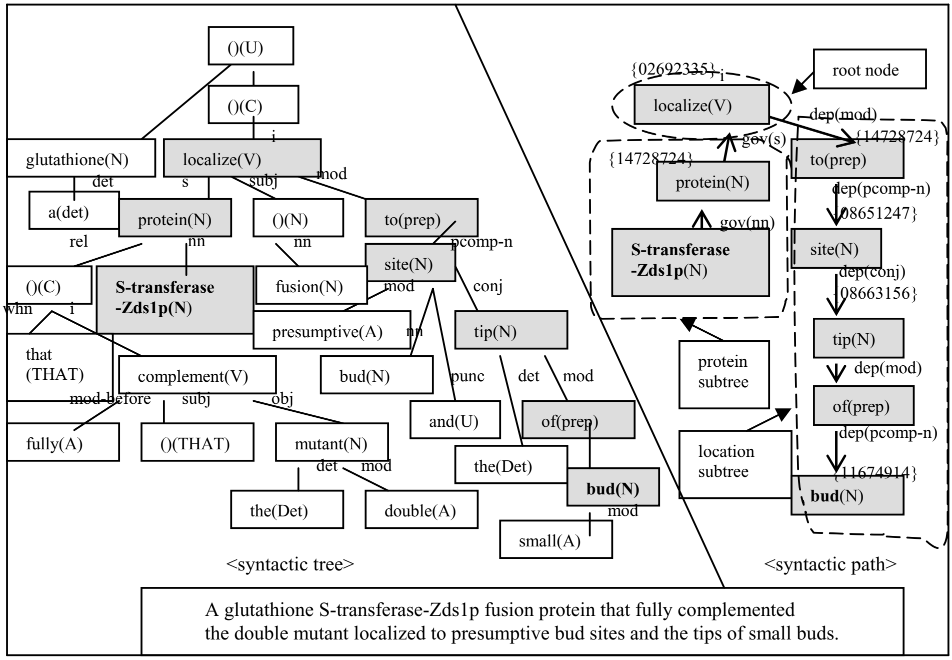


Figure 1. Example of a syntactic dependency tree and syntactic path from a protein node to its location node

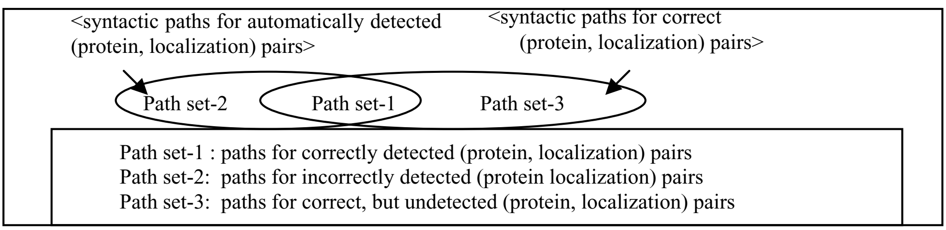


Figure 2. Three types of path sets

First, we construct a syntactic path for each (protein, localization) pair candidate in the training data. In the syntactic dependency tree of Figure 1, gray nodes indicate the path from the protein node '**S-transferase-Zds1p**' to the localization node '**bud'.** The right tree of Figure 1 shows the syntactic path for **(S-transferase-Zds1p, bud)**.

By comparing all automatically detected (protein, localization) pair candidates with correct (protein, localization) pairs in the training data, we construct three types of path set as shown in Figure 2. In the tree shown in Figure 1, localize(V) is the root. We call the left subtree of the root 'protein subtree'. In a similar way, we call the right subtree of the root 'location subtree'.

Table 1. Change in performance when any information is removed

Performance of test data	Using All information	Without any information		
		Without syntactic information		Without semantic information
		Without syntactic direction information	Without syntactic category information	Without synset offset
Precision	**0.4823**	0.4078	0.3954	0.4731
Recall	**0.7158**	0.5474	0.7263	0.6947
F-measure	**0.5763**	0.4674	0.5121	0.5629

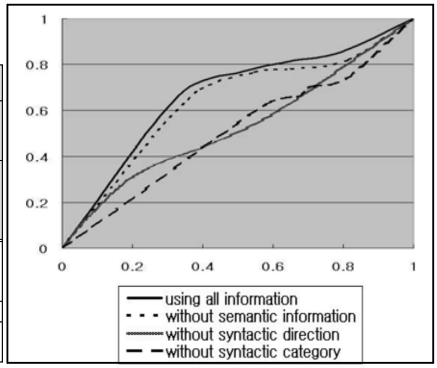


Figure 3. ROC curves for our experiments

The root node is an important key to connect between a protein subtree and a location subtree. So, we extract two types of template information from the constructed path set-1, -2, and -3: *i.e.*, <stem form of the root, syntactic category, weight> and <synset offset, syntactic category, weight >. Weight is the count of the template in the training data. We retrieve positive and negative root information. Positive root information is obtained from the correct path sets – path set-1 and set-3–, and negative root information is obtained from the negative path set – path set-2.

We retrieve syn-semantic patterns—*i.e.*, patterns consisting of syntactic and semantic information — for each node in subtrees from the three types of paths. For the syn-semantic patterns, we extract two types of template information: *i.e.*, <syntactic category of a node, stem form, syntactic direction, weight> and <syntactic category, synset offset, syntactic direction, weight>. We construct positive and negative syn-semantic patterns for subtrees.

We apply the obtained information to the test data as follows. For each syntactic path of the test set, if the weights of the negative root(syn-semantic pattern) templates matching the root(syn-semantic pattern) information of a syntactic path are more than those of the positive root(syn-semantic pattern) templates matching the root(syn-semantic pattern) information of the path, then we exclude the (protein, localization) pair candidate in the syntactic path from the protein localization results.

4. Experimental Results

For reasonable comparison with previous methods, we apply the data of Goadrich *et al.*[8] consisting of 7,245 sentences from 871 abstracts in the Medline abstract database. We use ten fold cross-validation for the experiments.

The experimental results are as follows:

1. Our proposed method for the detection of protein subcellular localization information achieved an F-measure of 0.5763 (Table 1).
2. When syntactic direction information is removed, the recall is decreased significantly (Table 1).
3. When syntactic category information is removed, the precision is decreased significantly (Table 1).

4. When semantic information is removed, the F-measure is slightly decreased (Table 1).
5. Our method outperforms previous method[8] of which the F-measure was 0.47.

Figure 3 shows receiver operating characteristic(ROC) curves of our experiments. Here, the true positive rate is plotted on the y-axis, and the false positive rate on the x-axis. As described in Figure 3, the experiment using all information has the highest and leftmost curve, showing that the performance of using all information is best.

5. Conclusion

To improve performance in detecting protein subcellular localization information, this paper proposes a method based on a syntactic dependency tree and WordNet thesaurus. First, we construct syntactic paths from a protein to its location candidate. Then, we retrieve positive and negative root information, and syn-semantic patterns for the protein subtrees and location subtrees. We detect (protein, localization) pairs according to the root information and syn-semantic patterns for subtrees. The experimental results have indicated that our method shows good performance with precision of 0.4823, recall of 0.7158, and F-measure of 0.5763. We have concluded that the proposed method is effective for detecting protein localization information. Furthermore, we have demonstrated that syntactic and semantic information is important for the performance of this method.

Acknowledgements

This work was supported by the Sungshin Women's University Research Grant of 2008.

References

[1] Rinaldi F, Schneider G, Kaljurand K, Hess M, Andronis C, Konstanti O, Persidis A, "Mining of Functional Relations between Genes and Proteins over Biomedical Scientific Literature using a Deep-Linguistic Approach", *Artificial Intelligence in Medicine,* Vol.39, Issue 2, pp.127-136, 2007

[2] S. Riedel, E. Klein, "Genic interaction extraction with semantic and syntactic chains", In *Proceedings of ICML05 Workshop on Learning Language in Logic* (LLL05), 2005

[3] B. Stapley, L. Kelley, M. Sternberg, "Predicting the sub-cellular location of proteins from text using support vector machines", *Proceedings of the Pacific Symposium on Biocomputing*, pp. 374-385, 2002

[4] H. Shatkay, A. Höglund, S. Brady, T. Blum, P. Dönnes, O. Kohlbacher, "SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data", *Bioinformatics* 23(11): pp. 1410-1417, 2007

[5] M. Craven and J. Kumlien, "Constructing Biological Knowledge Bases by Extracting Information from Text Sources," *Proc. 7th Int'l Conf. Intelligent Systems for Molecular Biology,* AAAI Press, pp. 77-86, 1999

[6] M. Skounakis, M. Craven, and S. Ray, "Hierarchical Hidden Markov Models for Information Extraction", *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 427-433, 2003

[7] D. Page and M. Craven, "Biological Applications of Multi-Relational Data Mining", *ACM SIGKDD Explorations Newsletter*, Vol.5, pp. 69-79, 2003

[8] M. Goadrich, L. Oliphant, and J. Shavlik, "Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction", *Proc. of the 14th International Conference on Inductive Logic Programming* (ILP), pp:98-115, 2004

[9] D. Lin, "Dependency-based evaluation of MINIPAR", In *Workshop on the Evaluation of Parsing Systems*, 1998

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-378

Capturing Knowledge Worker Behavior Based on Information Diffusion Theory

Masayuki Ishikawa [a,1], Peter Geczy [b], Noriaki Izumi [b], Takahira Yamaguchi [a]
[a] *Keio University, 3-14-1 Hiyoshi Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan*
[b] *National Institute of Advanced Industrial Science and Technology (AIST)*

Abstract. By analyzing the information diffusion in a studied large scale corporate portal, we discovered that only small population of users is sufficient to initiate the widespread of innovation. Applying these findings, we conducted analytic experiment for novel recommendation algorithm based on the browsing features of identified selected users and discovered that the first 10 users accessing a particular page play the key role in the information spread.

Keywords. Knowledge management technology, Collaborative filtering, Recommender System, Information Diffusion, Innovator theory

1. Introduction

Elucidation of human dynamics in electronic environments has been attracting significant attention in academic and commercial spheres [1]. Former experience of other users, expert opinions, and supplementary information sources help us decrease the level of uncertainty in purchasing decisions. Recommender system [2] using collaborative filtering [3] is an automated system providing suggestions to the users using activity decision of like-minded users. This approach is one of the effective systems for supporting decision making in electronic space. However, when we are dealing with community portal such as intranet, where novelty of the information are important, the activity decision of like-minded users may not be the most appropriate decision to adopt.

The data used in this work is a one year period intranet web log data of The National Institute of Advanced Industrial Science and Technology [5]. The majority of users are skilled knowledge workers. The intranet services support managerial, administrative and accounting processes, research cooperation with industry and other institutes, databases of research achievements, resource localization and search, attendance verification, and also numerous bulletin boards and document downloads. The information was structured according to the originating IP address, complete URL, base URL, script parameters, date-time stamp, source identification, and basic statistics.

[1]Corresponding Author: Masayuki Ishikawa, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan; E-mail:sekisen@ae.keio.ac.jp

Table 1. Information on data used in the study.

Data Volume	~60 GB
Average Daily Data Volume	~54 MB
Time Period	April 2005 – March 2006
Log Records	315,005,952
Unique URLs	3,015,848
Unique IP Addresses	22,704

The original innovator theory was presented in early sixties by Everett M. Rogers in his book "Diffusion of Innovations" [4]. Rogers defined the diffusion of an innovation as "the process by which an innovation is communicated through certain channels over time among the members of a social system". He pointed out that most innovations spread through the society via process characterized by an S-shaped curve formed by the cumulative frequency distribution of diffusion (diffusion curve). However the changing communication channel highly influenced by the widespread of internet will obviously speed up the flow and dissemination of information among the member of particular society. To examine the variability of innovation diffusion curves in the intranet, we randomly selected two thousand pages from the studied intranet portal that had more than thirty unique users accessing them. The diffusion curves were identified from the cumulative plots of the temporal access characteristics.

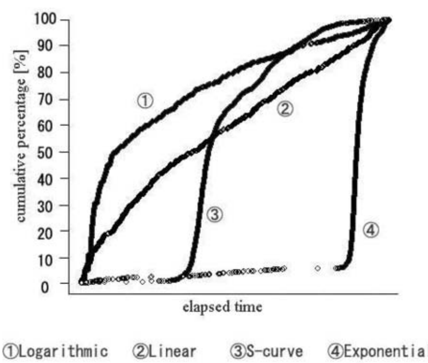


Figure 3. Various types of innovation diffusion curves.

Table 2. Percentual amount of the observed curve types in the studied corpus.

Diffusion Curve Type	Percentage [%]
Logarithmic	59.95
Linear	33.45
S-curve	3.3
Exponential	1.7
Others	1.6

It has been observed that the innovation diffusion curves can be categorized into the following four types: logarithmic, linear, S-curve, and exponential (see displays in Figure 3). The biggest category type of innovation diffusion curves is the logarithmic; approximately 60% (see Table 2). This indicates that innovations in the web environments are being accepted relatively rapidly. Only small population of users is sufficient to initiate the widespread of innovation. This suggests that by identifying the innovators, early adopters, and their visited pages, we can provide suggestions about the new and recent pages.

2. Proof of concept

Our analytic experiment aims to verify the result when the recommendations were provided according to the access log of innovator (first 2.5% of all users accessing the page), early adapter (next 13.5% of all user accessing the page) and first 10 users accessing the page. The experiment was run using the access data extracted from the server-side web logs of a large corporate intranet portal. Using randomly selected 2000 pages, we first identified reference users of the following three types: innovators, early adapters, and first 10 users that accessed the pages. Next, for each of the user X we are attempting to offer recommendation to, we identified a list of pages that the identified reference users have accessed during the one month prior to the time-stamp when user X accessed the particular page. The obtained list of pages is the initial potential recommendation list. We eliminated the pages that user X has already accessed, and evaluated and ranked the remaining pages. Simple evaluation and ranking is done with respect to the weighted access frequencies of the pages by reference users. The evaluated and ranked pages are then sorted and the 10 highest ranking page candidates is recommended to user X. Recommendation is determined correct if user X has accessed the page in the recommendation list within 3 month after the time-stamp when user X accessed a page.

2.1. Impact of Innovator, Early adapter, first 10 users on Recommendation Correctness

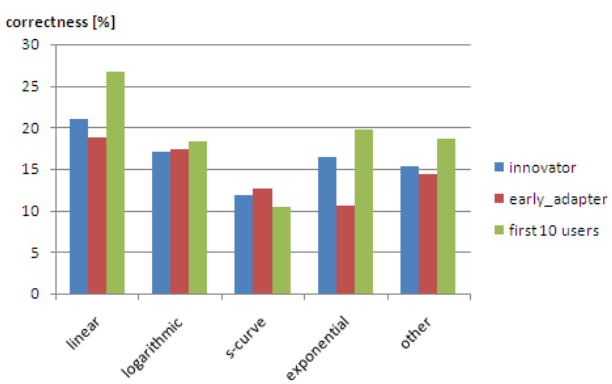


Figure 4. Difference in recommendation correctness when system recommends10 items

The former elucidation of user acceptance of innovations in web environments indicates that there are only few users that observe/discover the novelty at relatively very early stage, and the information about it propagates fast through the modern communication channels to the remaining relevant user populations. Figure 4 supports this assumption with the result that the first 10 users have higher correctness compared to innovator and early adapter in all cases except for s-curve.

2.2. Novel Recommendation Concept utilizing First 10 users

Simulation results indicate that the new and/or updated pages were recommended more frequently than the older pages. Figure 5 illustrates the relationship between how many times a page has been accessed before the recommendation, and the ratio between number of times accessed before recommendation and the total number of times recommended correctly (total of 4357619 correct recommendations). The curves have similar shapes for innovators, early adapters and first 10 users with the peak at pages accessed 200 or 300 times. This result indicates that the new and/or updated pages were recommended more frequently than the older pages. The difference is evident in pages accessed less than 100 times with the difference of 1% in correctness between first 10 users, innovators and early adapters. This clearly indicates that innovator and first 10 users could recommend new or recent items.

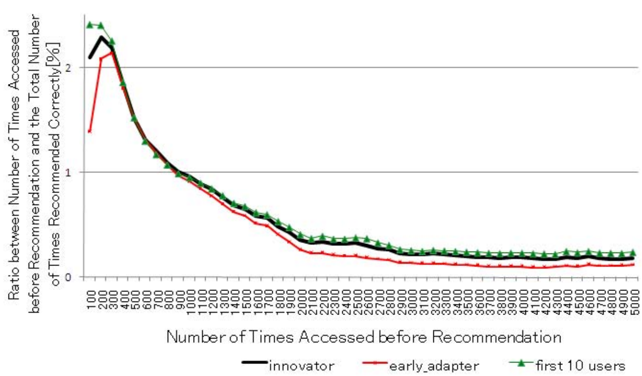


Figure 5. Ratio between number of times accessed before recommendation and the total number of times recommended correctly[%]

The results shown in Table3 further support the former elucidation, that innovator and first 10 users are suited for new item recommendation. Pages that have been accessed only once have been on average recommended 8 times for first 10 users and 34 times for innovator. With the increasing access history of a page, the number of recommendations to the page increases dramatically for first 10 users and innovator. Early adapter could not recommend pages with access history length of 1 and 2. Also the highest number of recommended times remained low; only 63 times at 10 access

history length. Comparing first 10 users and innovator, the pages with an access history length of 9 have been recommended over 900 times for first 10 users and 580 for innovator. This indicates that the group that is best suited for new item recommendation is first 10 users which support rapid innovation spread.

Table 3. Relation between access history length and times recommended

Access History Length	Times Recommended		
	First 10 users	Innovator	Early adapter
1	8	34	0
2	74	133	0
3	222	205	2
4	393	302	7
5	661	373	10
6	724	445	19
7	836	536	25
8	869	557	41
9	933	580	61
10	896	599	63

3. Conclusion

We presented a novel approach for providing recommendation using collaborative filtering engine to users with slow information acquisition using activity decision of users with fast information acquisition. The approach is based on analytic results of information diffusion characteristics in web environments. We have observed that the biggest category type of innovation diffusion curves is the logarithmic. This indicates that relatively small number of innovators and early adopters have a significant influence on the information spread. Further experiment using web logs of a large corporate intranet portal indicates that the first 10 users accessing the page is most suited for new item recommendation.

4. References

[1] Z. Dezso, E. Almaas, A. Lukacs, B. Racz, I. Szakadat, A.-L. Barabasi. “Dynamics of information access on the web”, Physical Review, E73:066132(6), 2006.

[2] P. Resnick and H. Varian, “Recommender System”, Communication of the ACM, Vol.40, No.30, pp.56-58, 1997

[3] Goldberg,D., Nichols,D., Oki,B.M., Terry,D.,“Using Collaborative Filtering to Weave an Information Tapestry”, Communications of the ACM, Vol.35, No.12, pp.61-70, 1992

[4] E. M. Rogers, *Diffusion of Innovations*, The Free Press of Glencoe, New York, 1962.

[5] P. Geczy, S. Akaho, N. Izumi, K. Hasida, “Extraction and Analysis of Knowledge Worker Activities on Intranet”, Practical Aspects of Knowledge Management, Springer-Verlag, Heidelberg, pp.73-85, 2006.]

Knowledge-Based Methods and Tools for Testing, Verification and Validation, Maintenance and Evolution

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-385

Exploiting Semantics in Collaborative Software Development Tasks

Dimitris PANAGIOTOU [a,1], Gregoris MENTZAS [a]
[a] *School of Electrical and Computer Engineering,, National Technical University of Athens ,Greece*

Abstract. Modern software development is highly knowledge intensive; it requires that software developers create and share new knowledge during their daily work. However, current software development environments are "syntantic", i.e. they do not facilitate understanding the semantics of software artifacts and hence cannot fully support the knowledge-driven activities of developers. In this paper we present SoWiSE, a Semantic Wiki environment which focuses on the software development domain and strives to address these problems. SoWiSE aims at providing software developers such a tool to ease their daily work and facilitate cooperation, context-aware navigation, ontology-based browsing and searching, concept-based source code documentation and related problem solving.

Keywords. Semantic Wikis, Software Development, Semantic Web, Collaborative Software Development

Introduction

Modern software development consists of typical knowledge intensive tasks, in the sense that it requires that software developers create and share new knowledge during their daily work. Although most software developers use modern state of the art tools, they still struggle with the use of technologies that are "syntantic", i.e. they use tools that do not facilitate the understanding of the concepts of the software artifacts they are managing (e.g. code fragments). Moreover since software development is a highly collaborative task, developers are in need of simple and easy-to-use tools that also enable collaborative work.

Collaboration and flexible ways of solving problems are necessary when a developer is frustrated investigating source code that he has never seen before (i.e. when extending a third party's software system) and is not capable of understanding its rationale. On the other hand, there are many situations that find a developer seeking source code that is already developed by others. He might not be aware of its existence, or even if he is, he is not able to find it effectively.

Most common practices used to resolve these problems include their discussion with other developers, finding relevant information on the Web, interacting with a dedicated forum, etc. But this kind of resolution is not unobtrusive and implies waste of

[1] Corresponding Author: Dimitris Panagiotou, NTUA, 9, Iroon Polytechniou str., Zografou Campus, Zografou, Athens 15780, Greece; E-mail: dpana@mail.ntua.gr.

valuable time. Sequentially, more adequate and unified means for working around these issues are required.

Wikis offer a fertile ground for investing on software development collaboration. These systems are gaining popularity and they provide flexible ways for articulating and sharing information among their users. But they suffer from information overload as pages written in a Wiki are unstructured and hard to retrieve at a later point of time.

On the other hand, recent trends in data integration and combination have led to the Semantic Web vision[2]. The Semantic Web strives to provide integration among information scattered across different locations in a uniform fashion. It defines a "common language" between these information resources.

In this paper we propose a new system – code-named SoWiSE for Semantic Wiki for Software Development – which is the result of exploiting the linkage of the Wiki paradigm and Semantic Web technologies in order to provide means for better collaboration in the software development domain and strive against the aforementioned problems. SoWiSE aims at providing software developers such a tool to ease their daily work and facilitate cooperation, source code documentation and problem solving.

The rest of the paper is organized as follows. In the next section, we describe what motivated us in developing the SoWiSE system. Afterwards, we describe the SoWiSE system in detail and then give the results of its experimental evaluation. Thereafter, we refer to related work performed in the field of our interest and state our contribution to it. Finally, we conclude the paper indicating our thoughts for future work.

1. Motivation

Herbsleb and Grinter in [1] identify that under time pressure to build a software system, the developers proceeded with coding, and slowly the code diverged from design. However other people were still relying on that documentation to design their own components and test suites. Over time, and especially in integration, all these inconsistencies came to light. Testers pointed to documentation as demonstration of why code was failing certain tests, other developers pointed to documents that described behavior that they had assumed was still exhibited by the code. All of these inconsistencies had to be resolved as part of building a working product.

Additionally, Herbsleb et al. in [2] argue that collaborations over distance must contend with the loss of the rich, subtle interactions that co-located teams use to coordinate their work. Previous research has suggested that one consequence of this loss is that cross-site work will take longer than comparable single-site work. In contrast to the rich interaction of face-to-face work, there is very convincing evidence that the frequency of communication drops off sharply with physical separation among co-workers' offices, and that the sphere of frequent communication is surprisingly small. Tom Allen [3], in a study of engineering organizations, reported that the frequency of communication among engineers decreased with distance. Further, he noted that when engineers' offices were about 30 meters or more apart, the frequency of communication dropped to nearly the same low level as people with offices separated by many miles.

[2] http://www.hpl.hp.com/semweb/sw-vision.htm

On the other hand these collaboration difficulties can be tackled using the Wiki paradigm. The success and great evolution of Wikis were based on their ease of use and significant flexibility they provided in supporting collaboration across a team of users or experts. Common usage of Wikis included project communication, intranets and documentation in enterprise settings. Nowadays, they are even more widely used by simple internet users who just need means to collaborate or exchange opinions with each other. A representative example of the "Wiki evolution" is Wikipedia[3].

Although Wikipedia's success is unambiguous it still has some drawbacks. A large side effect of such a large Wiki is that the user can be frustrated when searching for information. Since the largest percentage of the information in a Wiki is textual, the only way to find information is through a keyword-based search. This side effect is the result of unstructured accumulated information, and hence, it poses problems for knowledge management and productivity. The only semantics of Wiki pages lies in the links between pages. Most Wiki engines generate navigational structures from these links: one can see all pages linking to the current one, and go to these related pages. But this navigation through related pages is quite limited, and does not address the need for more advanced information retrieval [4].

The strong presence and yet promising research area of the Semantic Web is exactly the driving wheel behind our motivation in exploiting and applying these innovative technologies in the software development domain in order to boost a rather absent Semantic Wiki playground. The development of the SoWiSE system aims at supporting software developers when they collaborate in order to achieve their tasks in a more productive way.

2. SoWiSE

As mentioned above, the purpose of our work is to provide assistance to software developers in accomplishing their tasks in a more flexible fashion and shorten their total effort in terms of time. In order to achieve this goal, we approach software development documentation and related problem solving by combining lightweight yet powerful Wiki technologies with Semantic Web standards.

We chose Wikis for their significant support in knowledge articulation by providing lightweight and flexible mechanisms as well as semantics in order to add structured formulation to the Wiki contents. This combination gives software developers the opportunity to capture their knowledge related with a software development artifact exploiting a structured formalism. This knowledge will be then reusable, more easily shared and retrieved in the future by the Semantic Wiki's resilient user interface.

Finally, we decided to integrate SoWiSE seamlessly into the Eclipse IDE and deploy it as a plug-in for two reasons. The first reason is that we wanted to provide a proof of concept and chose Eclipse as a widely used and well known IDE which has also one of the broadest communities in the software development domain. Secondly, it was our intention not to change the working fashion of the software developer but instead to provide a tool environment that he is already familiar with and does not influence the way he is performing his tasks.

[3] http://www.wikipedia.org/

2.1. Software development ontologies

In order to ensure SoWiSE's effectiveness we have used it with a set of software development ontologies that were developed in the context of TEAM [5], which also constitute the initial set of SoWiSE's core ontologies. These ontologies can be extended by the SoWiSE system adding new concepts (classes). It is not intended to let the developer fully extend the ontology by defining relations and new properties as this would require basic knowledge and understanding of ontology engineering. However, the developer is free to extend the ontologies using any standard ontology editor such as Protégé [6].

These ontologies [7] combine content, interaction and context information to provide better support for knowledge sharing and reuse. They describe the structure and content of knowledge artifacts, their usage and organization and provide a basis for determining how artifacts are related. Different layers model different types of information: (a) the context layer describes people, various roles they assume within the organization, the projects they are involved in, teams performing the tasks, etc., (b) the interaction layer describes the structure of interaction (i.e. communication and collaboration) around bug reports, component reuse, discussion etc. and (c) the content layer determines the location, structure, content of software objects and related problem reports as well as the solution for problems.

The content ontologies used in SoWiSE are of crucial importance due to the fact that these are mainly the way to represent software development documents (e.g. a Java source code file). The content layer is further sub-divided into several ontologies, since they are sufficiently different domains, each meriting its own ontology for clarity. Each of these ontologies is relatively modular and is consequently usable for the most part independently of each other. The content layer consists of the following ontologies: (a) the software artifact ontology that describes different types of software artifacts such as the structure of the project source code, the components to be reused and software documentation, (b) the problem/solution ontology that describes the problems that may arise during coding (e.g. submitted bug and patch reports and their attributes) and how they are resolved, (c) the lifecycle ontology that describes decision made during programming or resolving problems and (d) the vocabulary that describes general software development terminology.

2.2. Architecture

Figure 1 depicts a simplified version of the Eclipse SDK architecture [8] and how the SoWiSE system is built on top of the EclipseWiki plug-in[4] using the Eclipse Plug-in Development Environment (PDE) [8].

An Eclipse plug-in is the smallest atomic unit of the Eclipse Platform [8]. It is a common practice for software developers to build separate plug-ins and compose more complex tools out of them. All of the Eclipse Platform's functionality is based on plug-ins apart from a small kernel known as the Platform Runtime. Each plug-in declares several named extension points and optionally extends one or more plug-ins using in

[4] http://sourceforge.net/projects/eclipsewiki/

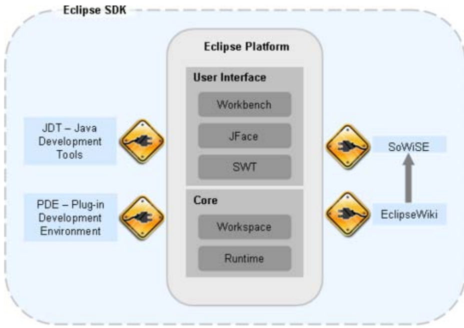


Figure 1. SoWiSE interconnection with Eclipse SDK and EclipseWiki

turn their extension points. This plug-in extension mechanism is the heart of the Eclipse PDE.

Based on this concept, we built the SoWiSE system exploiting well known and tested functionality of the Eclipse Platform such as SWT widgets, JFace, core runtime operations etc. On top of that, we needed some basic conventional Wiki functionality (non-semantic Wiki) in order to concentrate on building the semantic aspects of the Wiki and customize it for our specific domain: software development. EclipseWiki provides such basic functionality and it is available as an open-source Eclipse plug-in download.

The SoWiSE editor is developed as an Eclipse plug-in providing seamless integration with the IDE and is based on the Rich Client Platform (RCP) [8]. It takes advantage of out-of-the-box functionality of SWT widgets and JFace components for its GUI namely toolbars, buttons, user dialogs, text editors and browsers.

Additionally, it extends the basic functionality of EclipseWiki and provides more sophisticated means for creating and manipulating Wiki pages. During the effort of customizing EclipseWiki for our purposes, many pieces of code of the original EclipseWiki plug-in were modified. For instance, new color coding schemas for Java classes[5] were defined and implemented by changing the existing code of EclipseWiki (Figure 2b). On the contrary, extending EclipseWiki required new Java classes to be implemented – the new action toolbar and Wiki page statistics which are realized as an individual Eclipse view is such an example (Figure 2c).

2.3. Features and usage

EclipseWiki already provided us among other features the following: (1) linking to local Wiki documents, Eclipse workspace resources, Java source code and urls, (2) definition of common prefixes to websites, Wikis and Eclipse resources using a local WikiSpace, (3) export of the user's Wiki, (4) multipage editor with source, browser (TWiki[6], SnipSnap[7] and Ward's Wiki[8] renderers) and syntax views, (5) headers and footers in the browser view and exported documents, (6) support for embedding other

[5] SoWiSE can manipulate source code of whatever programming language, but it is currently optimized for use with Java.

[6] http://twiki.org/

[7] http://snipsnap.org/

[8] http://c2.com/cgi/wiki

Wiki documents, eclipse resources and Java code in Wiki documents, (7) syntax coloring for links and (8) outline of document headings.

Some of the above features were customized and new ones were realized. The following describes customized and new supported features by SoWiSE as well as their usage.

2.3.1. Linking

Apart from EclipseWiki's capabilities for linking, SoWiSE provides statistical information for each Wiki page displaying "existing links" and "unknown links". Existing links represent how many Wiki pages were already created "automatically" while unknown links represent how many Wiki pages could be created automatically. Automatically, in this context means that SoWiSE recognizes words which match with a specific pattern and gives the possibility to the user of creating a new Wiki page (by clicking on a blue question mark, Figure 2b) and a link to it simultaneously.

2.3.2. Multipage editor

We have currently customized EclipseWiki's Ward's Wiki renderer tweaking mainly its appearance. This renderer is the default browser for SoWiSE. The browser's action toolbar was redesigned providing a clearer and more elegant interaction point with the user. Different font styles were defined for rendering different parts of the Wiki page depending on the type of the rendered source code (e.g. import statements, comments, etc.). We are planning to add support for TWiki and SnipSnap syntaxes customizing the respective renderers of EclipseWiki.

The source editor of EclipseWiki is an extension of a standard Eclipse text editor which provides a complete set of common text editing functionalities. In this editor the user is able to compose a Wiki page usually by embedding a Java class file, documenting it using the Wiki's syntax, attaching related resources and by tagging (see below) important terms that are classified in an ontology schema. In order to support tagging and attachments, customization of the original source editor of EclipseWiki was required.

On top of that, the customized editor is seamlessly integrated into the Eclipse IDE (being part of an Eclipse plug-in as mentioned above) and thus provides support for features such as: management of Wiki page files using standard Eclipse navigation views and menus (save, open etc.), advanced text find/replace mechanisms, cut/copy/paste facilities and so on. Finally, the multipage editor has a syntax view where the user can have an overview of the Wiki's syntax rules.

2.3.3. Search

Search in SoWiSE is twofold. The user is able to search within a Wiki page using the typical search mechanism of Eclipse mentioned above, as well as to perform a search across the whole local WikiSpace. The search button in the action toolbar serves exactly this last option. It enables the user to perform an ontology-based query and find Wiki pages that are tagged with a specific term given by the user. The term is specified either as keyword(s) that the user remembers and is capable of filling out or by navigation in the ontology tree of the local WikiSpace.

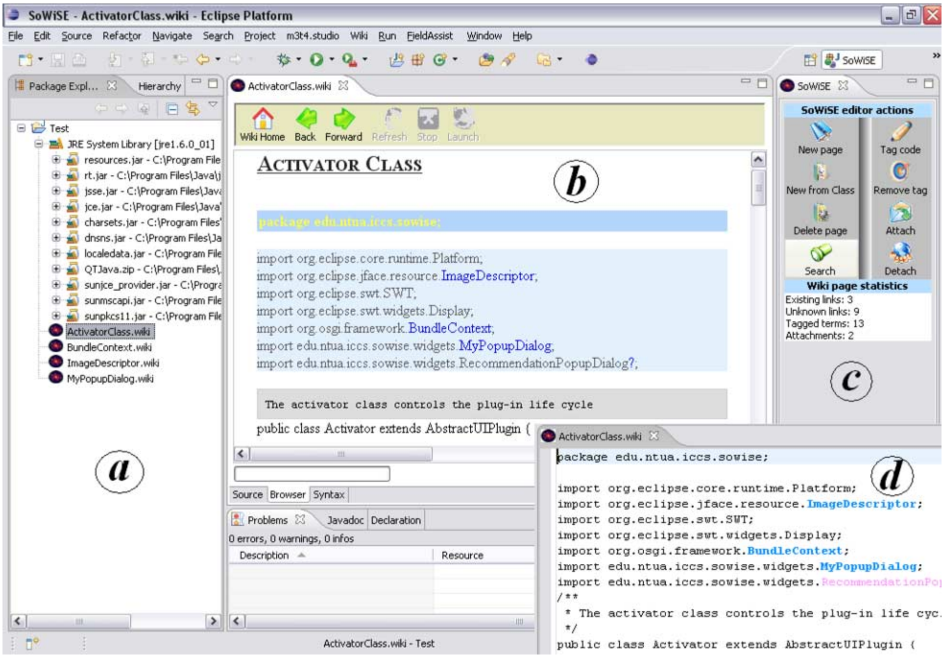


Figure 2. SoWiSE graphical user interface: a) Eclipse Package Explorer view, b) SoWiSE browser in multipage editor view, c) SoWiSE action toolbar and Wiki page statistics view, d) SoWiSE source editor

An important aspect of the search mechanism is the use of inference. SoWiSE takes advantage of knowledge that is inferred by applying rules specified by the user. For example, if the user declares that an individual software developer is an expert in SWT, when the user actually searches for information related to SWT he is presented with all Wiki pages referring to SWT as well as related resources that link him to the expert.

2.3.4. Tagging

Tagged terms are easily identified in the source or browser views by a red underline symbolism. When the user hovers the mouse above the underlined term then a small popup dialog appears displaying the corresponding ontology concept and a small part of the ontology tree nearby that concept. Tagging is feasible only by highlighting a term in the source view of SoWiSE and pressing the "Tag code" button. The number of the tagged terms in a Wiki page is displayed in the Wiki page statistics part of the SoWiSE plug-in. Removal of a tag is accomplished by positioning the cursor over a tagged term and clicking on the "Remove tag" button or alternatively using SoWiSE's context menu.

2.3.5. Attachments

The attachment mechanism of SoWiSE is used to relate a specific excerpt of a Wiki page with an information resource. The actual document is not attached persistently on the Wiki page but rather referenced by it. For example, this facilitates a developer

when looking at a piece of code in order to be able to find further related information and help about the usage or usefulness of that specific segment. Related resources could be of any electronic format: email messages, web resources, office documents, images, videos etc. At the same time, we are even exploring whether it is feasible to realize a model for associating “physical” resources such as a book with these Wiki page excerpts.

Performing attaching and detaching of resources using the corresponding buttons of the user interface follows the same approach of the tag/remove tag buttons. The number of attachments of the Wiki page is displayed in the statistics as well. Attached resources are identified in the Wiki page by clip symbols which are in fact active links to those resources. Similarly, as in the case of tagged terms, the excerpts that have one or more attached resources are surrounded by red parenthesis.

3. Experimental Evaluation

We have not performed a comprehensive assessment yet, but we have deployed SoWiSE in small teams of software developers consisting of four to six persons in the following organizations: (1) a Brussels based company specializing in the field of Information and Communication Technology (ICT) services (Intrasoft International S.A.), (2) a leading hungarian association dealing with open source software at corporate level (Linux Industrial Association) and (3) an italian company which operates in the Information Technology market, focusing on business applications (TXT e-Solutions). The feedback that we have received so far is positive and the fact that almost every software developer considers SoWiSE as an important tool in assisting him in his tasks is very promising.

More specifically, all of the users agree that SoWiSE’s GUI is easy to use and straightforward. They are satisfied by the fact that it is seamlessly integrated into the Eclipse IDE and they do not have to launch an external tool or change the way they work in their development environment. The convenient toolbar and the easiness of adding annotations to the source code are also well accepted by most of the users. Additionally, the way for finding relevant information or possible solutions to problems through the Wiki using the search facility of SoWiSE is more than welcome.

On the other hand, a significant percentage of the users would desire a more flexible way for tagging source code, namely semi-automatic generation of annotations in order to reduce the burden of manually created annotations. Thus, their role of being human annotators would change to supervisors saving valuable time. Another important issue according to the users is navigation. They consider that alternative means for navigation should be provided, including more intuitive representation of the Wiki’s contents such as faceted browsing, different views over the same Wiki page, etc.

4. Related Work

Besides general purpose Semantic Wikis, some domain specific implementations exist. In our context we have found out two implementations in a relative and broadest domain than software development: software engineering. Following is a brief description of these systems.

RISE – Reuse in Software Engineering [9]: The RISE project introduces the Riki Wiki combined with the Wikitology paradigm [10, 11]. With Wikitology, it is possible to derive semi-automatically an ontology needed for information retrieval from the pages in the Riki itself. Such a Wikitology automatically updates "itself", i.e., it reflects the changes of contained knowledge, changed views of the users accounts, new projects, customers, rules, trends, and ideas inside the Wiki. By considering the Riki as the ontology, the ontology will be always collectively edited up-to-date and will automatically reflect all changes.

Hyena: Hyena [12] aims at providing a platform that makes RDF's advantages available to software engineering without diverging too far from the concepts and tools that are familiar in that domain. It provides editing, GUI and publishing services for implementing editors of RDF-encoded custom models and it is implemented as an Eclipse plug-in. It includes some built-in ontologies for many basic software engineering tasks. These ontologies have themselves become part of the platform, because they provide services that will be useful for other ontologies.

5. Our Contribution

SoWiSE contributes in the Semantic Wiki sphere mainly at the software development domain rather than just providing another Semantic Wiki alternative. We feel that in this specific domain the software developer is confronting a lack of tools and options in supporting him to accomplish his task. On the other hand, software developers need not only tools, but in order for them to be helpful and successful these must be convenient and require little training effort. Hyena, for example, is too much complicated even for software developers providing a few sophisticated ontology editors and forcing the user to acquire the basic knowledge of an ontology engineer.

We aim at providing support to the software developer for documenting code, finding the best solution to a problem or the way to move forward in critical decisions in order to save time during the heavy task of software development. Of course, as described above this support should require minimum effort for learning SoWiSE which should also be integrated seamlessly into the software developer's environment. For these reasons, we chose to develop SoWiSE as an Eclipse plug-in as a proof of concept and to integrate it seamlessly to the Eclipse IDE. On top of that, we designed and implemented a rather familiar and easy to use GUI in order to give the opportunity to software developers of just learning SoWiSE's capabilities and not wasting time on learning its user interface.

Another important contribution of SoWiSE is the fact that it enables tagging of specific fragments of a Wiki page which is an immediate issue in current research. Most existing Semantic Wikis take the simple approach of having one Wiki page per concept. But this is not convenient for our specific domain, as in software development there is a clear discrimination between different fragments of a specific source code file which is documented as a Wiki page, for example. Thus, it is apparent that this simplification does not hold true in our case, having in mind that the software developer needs to tag different source code fragments in the same Wiki page. As Kiesel notes in [13] "This simplification makes it almost as easy to express semantic statements about a concept as it is to make a link to a Wiki page. However, this simplified model puts a restriction on the level of granularity of possible annotations: if

you can only refer to “the concept of the page”, how do you refer to a specific fragment of the page?”

6. Conclusion and Further Work

In this paper we presented SoWiSE, a Semantic Wiki for assisting software developers to cooperate and accomplish their tasks in a more flexible manner. The lightweight approach of the Wiki paradigm for documentation and problem solving is further facilitated by using Semantic Web technologies. We have evaluated the system in three large organizations and the resulting feedback is positive – a fact that motivates us to further extend the system.

We plan to extend the system by providing even more intuitive facilities for articulating, navigating and retrieving the Wiki’s contents. A more formal and comprehensive assessment is planned and the results will have impact on the further development of SoWiSE.

Acknowledgements

This work was partly supported by the TEAM project, which is funded by the EU-IST program under grant FP6-35111. The authors are responsible for the content of this publication.

References

[1] Herbsleb, J. D., and Grinter, R. E. Splitting the organization and integrating the code: Conway's law revisited. In *Proc. of the 21st ICSE 1999*, IEEE Computer Society Press, Los Alamitos, CA, 85-95, 1999.
[2] Herbsleb, J. D., Mockus, A., Finholt, T. A., and Grinter, R. E. Distance, dependencies, and delay in a global collaboration. In *Proc. of the CSCW 2000*. ACM, New York, NY, 319-328, 2000.
[3] Allen, T. J., Managing the Flow of Technology. Cambridge, MA: MIT Press, 1977.
[4] Oren, E. SemperWiki: a semantic personal Wiki. *Proc. ISWC 2005*.
[5] TEAM - Tightening knowledge sharing in distributed software communities by applying semantic technologies. http://www.team-project.eu/.
[6] The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/.
[7] Georgousopoulos C., Happel H-J., Khan O., Maalej W., Narendula R., Ntioudis S., Panagiotou D., and Stojanovic L. TEAM Project Deliverable D9: User Requirements and Conceptual Architecture.
[8] Eclipse Platform Technical Overview. http://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html.
[9] Decker B., Ras E., Rech J., Klein B., and Hoecht C. Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis. *Proc. ISWC 2005*.
[10] Decker, B., Ras, E., Rech, J., Klein, B., Reuschling, C., Höcht, C., Kilian, L., Traphöner, R., and Haas, V. A Framework for Agile Reuse in Software Engineering using Wiki Technology. *Proc. PKM 2005*, p 411-414.
[11] Klein, B., Höcht, C., and Decker, B. Beyond Capturing and Maintaining Software Engineering Knowledge – “Wikitology” as Shared Semantics. *Proc. AI 2005*.
[12] Rauschmayer, A. An RDF Editing Platform for Software Engineering. *Proc. ISWC 2005*.
[13] Kiesel, M. Kaukolu: Hub of the Semantic Corporate Intranet. *Proc. ESWC 2006*.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-395

Usability evaluation of software applications with the use of Usability Logger

Nektarios KOSTARAS [a], Michalis XENOS [a,1] and Christos KOUTSOGIANNIS [a]
[a] *School of Sciences and Technology, Hellenic Open University, Patras, Greece.*

Abstract. This paper presents the main characteristics and capabilities of Usability Logger software tool, used in the usability evaluation of Human Computer interaction interfaces. Usability Logger can be employed to record the actions performed in workstations, with the use of the I/O devices. In this paper the tool is analyzed, compared with other similar tools and used in a case study involving the usability evaluation of two software systems according to predefined criteria.

Keywords. Software Quality, Usability Evaluation, Software Benchmarks

Introduction

Within the last years, a major shift has occurred in the development, design and deployment of software applications concerning usability. Nowadays usability is no longer a luxury, but rather a basic determinant of productivity and of the acceptance of software applications [1]. The usability of software applications is of great importance since it determines to which extent the provided features will be used by its users [2], [3]. Through time many definitions for usability have been published. Two of the most established definitions can be found in international standard for the evaluation of software ISO 9241-11 [4] and ISO 9126 [5]. ISO 9241-11 defines usability as "*the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*". In ISO 9126, usability is defined as "*the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions*".

As it becomes obvious, usability evaluation has turned out to be an essential process in the development of software applications. This process can be performed using established methods, suitably designed usability evaluation laboratories with specific equipment and various software tools. One of the usability evaluation tools that developed from the Software Quality Research Group of the Hellenic Open University is the *Usability Logger* [6]. *Usability Logger* can be integrated in a usability evaluation test and assist the process by recording the actions of a user while interacting with the interface of a software application. The tool records the use of the keyboard, the mouse and takes screenshots from the interaction of the user with the interface. A competitive feature of the *Usability Logger* is the easiness with which can be installed and used

[1] Corresponding author. 23 Sachtouri Str, GR 26222, Patras, Greece. Tel.: +30 2610 367405; fax: +30 2610 367416. E-mail address: xenos@eap.gr

either in a usability evaluation laboratory or in any other place were the software application under evaluation is used.

The rest of the paper is structured in 5 sections and presents the *Usability Logger* tool and its use. In section 1, a reference to usability evaluation software is made and a comparison among these tools including *Usability Logger* is performed. In section 2 the *Usability Logger* tool is presented. Section 3, describes a case study that includes the usability evaluation of two software applications with the employment of *Usability Logger*. Section 4 shows the results of the usability evaluation. Finally, the conclusions drawn from the use of the software tool in usability evaluation tests are discussed.

1. Usability Assessment software

There are various software tools getting involved in the usability evaluation process and which are similar to *Usability Logger*. Some of them are freeware, but most of them are commercial. Table 1 presents software similar to *Usability Logger*. In this table, the name and version of the software tested is mentioned as well as the website that can be found. In the last column of Table 1 there is an indication whether the software is distributed for free (V) or not (X).

Table 1. Usability assessment software

	Software Name	Website	Freeware
1.	Mousotron Pro 5.0	www.freedownloadscenter.com	V
2.	Mouse Off-road 2.15	www.tucows.com	V
3.	Mini-Input 2.0	www.filesrepository.com	X
4.	Mouse Odometer 4.0	www.introspectsoftware.com	V
5.	Mouse Meter 1.51	www.softplatz.com	X
6.	My Mouse Meter 1.0.9	www.softpedia.com	V
7.	Mouse Clocker 1.0	www.download.com	V
8.	Exact Mouse 2.0	www.softpedia.com	X
9.	Usability Logger 2.3	http://quality.eap.gr	V
10.	321Soft Screen Video Recorder 1.05	www.filedudes.com	X
11.	Screen VidShot 2.2.0.14	www.filesrepository.com	X
12.	ZD Soft Screen Recorder 2.6.4.0	www.softpedia.com	X
13.	Screen Video Recorder 1.5	www.filesland.com	X
14.	Screen Tracker 2.0	www.bestsware.com	X
15.	Advanced Key and Mouse Recorder 2.80	www.freedownloadscenter.com	X
16.	Action Mouse Mover 1.0	www.supershareware.com	X
17.	Adamant Key Mouse Pro 3.3	www.softpedia.com	X
18.	Axife Mouse Recorder 5.01	www.soft411.com	X
19.	ECTI 1.73	www.freedownloadscenter.com	X

20.	Mouse Tamer 2.0	www.supershareware.com	x
21.	Smack 1.06	www.filesrepository.com	x
22.	Mouse Machine 1.1	www.softpedia.com	v
23.	Jitbit Macro Recorder 3.82	www.filesrepository.com	x
24.	Mouse Master 2.1	www.tucows.com	x
25.	Macro Wizard 4.1	www.softpedia.com	x

The software listed in Table 1 was tested according to some basic features that are essential for the expected functionality of logging software and a comparative evaluation was produced. The functions took into consideration are:

1. Distance covered by mouse cursor
2. Use of keyboard
3. Use of mouse buttons
4. Total time of user activity
5. Storage of user activities in log files
6. Creation, storage and implementation of macros
7. Screen captures
8. Video captures
9. Speed of cursor

The results of the comparative evaluation are presented in Table 2, where with "v" is marked a function included in the software and with "x" not included.

Table 2. Comparison of usability evaluation software features

	Software Name	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1.	Mousotron Pro 5.0	v	v	v	v	v	x	x	x	v
2.	Mouse Off-road 2.15	v	x	x	v	x	x	x	x	v
3.	Mini-Input 2.0	v	v	v	x	v	x	x	x	x
4.	Mouse Odometer 4.0	v	v	v	x	v	x	x	x	x
5.	Mouse Meter 1.51	v	v	v	v	v	x	x	x	v
6.	My Mouse Meter 1.0.9	v	v	v	v	x	x	x	x	x
7.	Mouse Clocker 1.0	v	x	x	x	x	x	x	x	x
8.	Exact Mouse 2.0	v	x	x	x	x	x	x	x	x
9.	Usability Logger 2.3	v	v	v	v	v	x	v	x	x
10.	321Soft Screen Video Recorder 1.05	x	x	x	x	x	x	x	v	x
11.	Screen VidShot 2.2.0.14	x	x	x	x	x	x	v	v	x
12.	ZD Soft Screen Recorder 2.6.4.0	x	x	x	x	x	x	x	v	x
13.	Screen Video	x	x	x	v	x	x	x	v	x

	Recorder 1.5									
14.	Screen Tracker 2.0	X	X	X	V	X	X	V	X	X
15.	Advanced Key and Mouse Recorder 2.80	V	V	V	X	V	V	X	X	X
16.	Action Mouse Mover 1.0	X	X	X	X	V	V	X	X	X
17.	Adamant Key Mouse Pro 3.3	X	V	V	X	X	V	X	X	X
18.	Axife Mouse Recorder 5.01	V	V	V	V	X	V	X	X	X
19.	ECTI 1.73	V	X	V	V	X	V	V	X	X
20.	Mouse Tamer 2.0	V	V	V	X	X	V	X	X	X
21.	Smack 1.06	V	V	V	X	X	V	X	X	X
22.	Mouse Machine 1.1	V	X	V	X	X	V	X	X	X
23.	Jitbit Macro Recorder 3.82	V	V	V	X	X	V	X	X	X
24.	Mouse Master 2.1	V	V	V	V	X	V	X	X	X
25.	Macro Wizard 4.1	V	V	V	X	V	V	X	X	X

Considering Table 2, it is obvious that *Usability Logger* has the advantage that can record the use of the keyboard and mouse buttons, the distance covered by mouse cursor, the total time of user activity and store these data together with screenshots, from the interaction with the interface, in the database and display this information in one form. Furthermore, *Usability Logger* has a feature not mentioned in Table 2. It can measure the distance covered by the mouse cursor in specified time slots, thus calculating mouse movement for specific instances. This property is useful in quantitative evaluation.

2. The Usability Logger tool

As already mentioned, *Usability Logger* is a software tool that is used in usability evaluation process. It records the actions performed with the use of I/O devises like keyboard, mouse and screen, during the interaction of a user with the interface of a software system, in order to evaluate its usability. This evaluation can take place in various environments where the software application under evaluation is installed. Such a place can be a usability evaluation laboratory, or any other place where the application is used. The usability logger can be installed and used easily in every working environment. It can be also used in cases that usability evaluation tests include simultaneous use in more than one workstation. In such cases *Usability Logger* can be installed in as much workstations as needed, which belong in the same Local Area Network (LAN), start the recording of the I/O devices of each one at the same time with the others and store the data collected in a server that is situated in the LAN.

The current version of the software tool is *Usability Logger* v2.3 and in its interface the language used is Greek. In the future, it will be created a version including English interface.

Usability Logger comprises of two parts. The *Recording Tool* and the *Database* in which data collected from the recording tool is stored and analyzed.

2.1. Recording Tool

The *Recording Tool* is recording the movement and the distance covered by the cursor in the screen as well as the use of the mouse buttons and their combinations (e.g. double click), the buttons pressed in the keyboard and screenshots that reveal the actions of the user during the usability evaluation. All the functions mentioned in the previous paragraph are carried out in the background so as not to be visible by the user and interfere in the process. In order to start the observation**,** the main form of the *Recording Tool* that is presented in Figure 1 has to be completed. In this form the user selects the scenario which will execute and inserts the username and the password with which is registered in the database, in order to start the usability evaluation.

The recording of the screen, takes place in short time intervals which are selected according to the needs of the usability evaluation. The choice of the time intervals can be selected from a drop down menu in the *Recording Tool* interface, which has three choices as it is shown in Figure 1.

Important characteristic of the *Recording Tool* is that it can be installed and used at the same time in more than one workstation. This attribute gives the flexibility to run usability evaluation tests at the same time in a usability evaluation laboratory for example and collect the data of the recordings simultaneously in the database, which can be installed in a server within the local network.

2.2. Database

The second part of the *Usability Logger* tool is its *Database*. The *Database* is used to store information regarding the scenarios from usability evaluation tests, the users participated in each scenario and data from the actions of users during usability evaluation tests like recording of I/O devises.

The form of the *Database* that presents data relevant to the I/O devises is shown in Figure 2. In this form screenshots from the interaction with the workstation, for every time interval that we have selected, are presented and for every screenshot relevant information is presented.

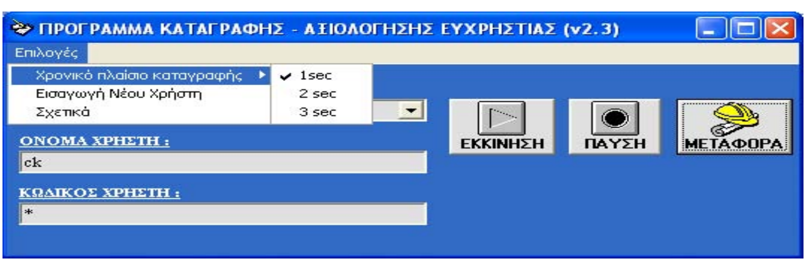


Figure 1: Starting form of the *Recording Tool* including time interval choice menu

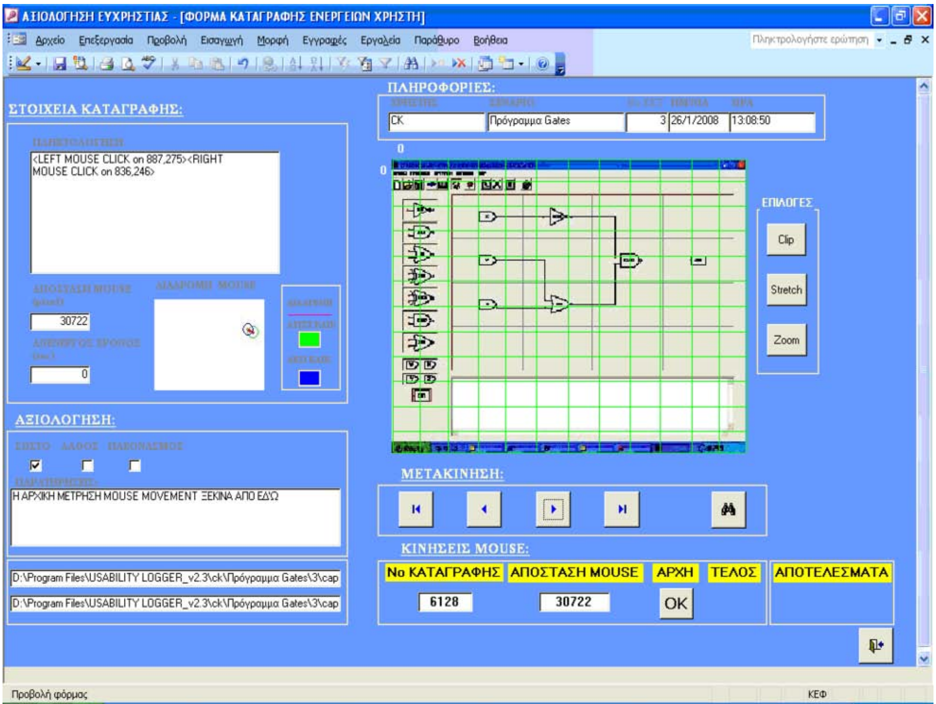


Figure 2: Data stored in the *Database* relevant to the I/O devices

As we can see in Figure 2, on the top right of the form there is information that presents the name of the user, the name and the number of the scenario performed in the usability test, the date that the test took place and the time started. On the top left of the form, information of the mouse clicks, the distance covered by its cursor and the idle time as well as a picture of the cursor movements are presented. Just below, the usability expert can add comments on every action taken during the usability evaluation. These comments can be made on a wrong or right action, if an action is not necessary and for any other reason that might be useful in the analysis of the results. Finally, on the bottom right corner there are buttons that control the presentation of the screenshots and just bellow that, there is the area that measurements concerning the distance covered by the mouse cursor for specified duration are recorded.

Using the database the usability evaluation expert can perform a number of actions which can be selected from the interface presented in Figure 3. Using this interface, usability evaluation expert can visualize the use of the I/O devises (Figure 2), export statistical data regarding the use of the I/O devises and especially the mouse movements create scenarios of usability evaluation tests and register users that participate in these tests. The database can be installed either locally in the workstation that is used for the evaluation or in a server that is part of a Local Area Network.

The screenshots that are collected during the recording of the users' actions are all stored in the *Database* and they are categorized in different files, depending on the user, the name of the evaluation scenario and the number of times each scenario is implemented.

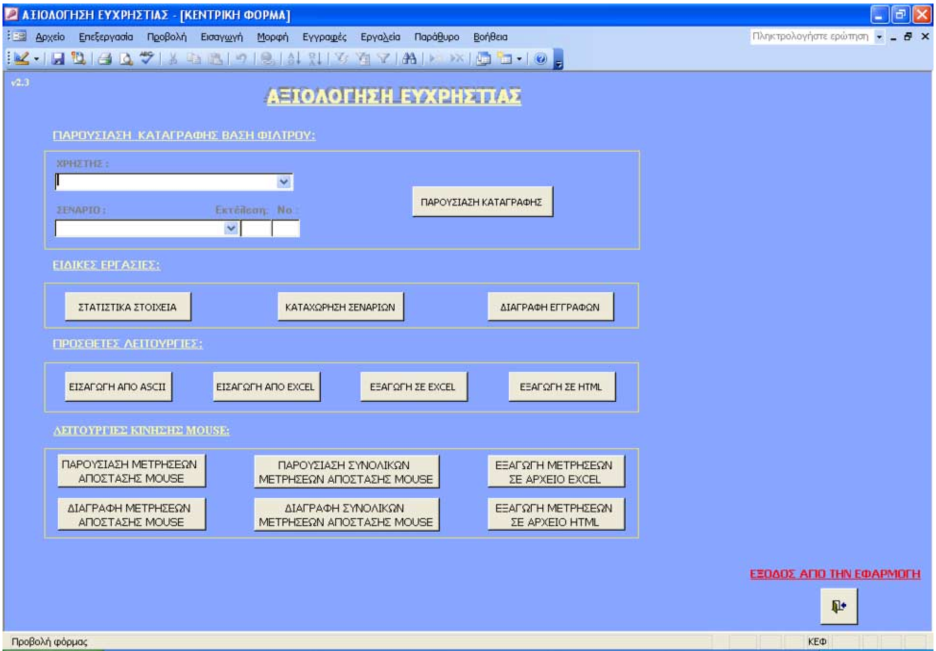


Figure 3: Main form of the *Database*

If the *Database* is installed in a server then the files created are maintained locally, in the workstation, during the time of the recording. At the end of the process these files are automatically erased from the workstation and transferred to the server. The screenshots are saved in jpeg format so as to keep the size of the file as small as possible, since after extensive use of the *recording tool* it can become very big.

3. Case studies of Usability Logger tool

In this section there will be a presentation of the utilization of *Usability Logger* in a usability evaluation process of the software applications "*GATES*" and "*Electronic Lite*" [7].

The usability evaluation of the two software applications mentioned earlier, took place in the Software Quality Assessment Laboratory (SQAL) of the Hellenic Open University. The SQAL is a suitably equipped laboratory, in which experiments concerning the quality of various software applications take place. This laboratory consists of two rooms, one designated as the experiment room, in which the users are sited and a second designated as an observation and control room, in which the usability experts are sited. The observation room is separated from the experiment room by one-way mirror so that evaluators can watch participants performing, but not vice versa.

The number of users participated in the usability evaluation was eight. In the beginning, participants received a short presentation introducing the software under evaluation, through a demo presentation. The purpose and objective of the experiment

was also presented, as well as some indications about what they are expected to do. Participants were also assured and reminded that the purpose of the evaluation is the usability of the software and not their personal performance. Hence, they were encouraged to act in a way that is typical and comfortable to them. Finally, participants were informed that they are being observed and all their actions are recorded.

After the orientation, the usability evaluation took place, during which the participants were asked to perform a number of predefined tasks. The scenario stood as follows:

- The participants were asked to enter the experiment room of the Software Quality Assessment Laboratory (one at a time) and sit down at a desk where a PC was placed with the software under evaluation installed.
- Participants were asked to start the software and perform a number of predefined tasks which were written in a piece of paper that was situated on the desk.
- After all tasks were completed the participant should leave the test room.

Each user participated in the usability evaluation of the two software applications. First was performing the evaluation of the "*GATES*" and then of the "*Electronic Lite*" software. "*GATES*" is educational software that helps students of Electronics to get a better understanding on "*gates*". "*Electronic Lite*" is educational software that can be used in high school courses related to Electrical and Electronic Engineering.

One of the tools used for the recording of the participants' actions was the *Usability Logger*. The two parts of the tool were installed in different locations. The *Recording Tool* was installed in the workstation that the participant was using for the interaction with the software and the *Database* was installed in a server placed in the control room of the laboratory. The *Recording Tool* was started each time a participant was beginning the interaction with the interface of the software under assessment and stopped when all the tasks were completed. With the completion of each scenario the *Recording Tool* was sending all the recorded data to the *Database* in order to be analyzed by the usability experts at the end of the entire process.

4. Results

At the end of the usability evaluation process described in section 3, usability evaluation experts studied the data collected. This data was stored in the *Database* that was installed in a server situated in the control room.

In the first face of the data analysis, usability experts went through the forms like the one presented in Figure 2. Using this type of forms they were focusing on specific aspects that were presented and which are mentioned in subsection 2.2.

In the second face they used the form presented in Figure 3, in order to get general statistics about the performance of all the participants in both experiments. In Figures 4 and 5 are presented Tables that show the actions of all users in the usability evaluation of "*GATES*" and "*Electronic Lite*" software respectively.

ΕΜΦΑΝΙΣΗ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΓΙΑ ΤΟ ΣΕΝΑΡΙΟ Πρόγραμμα Gates

ΟΝΟΜΑ ΧΡΗΣΤΗ	No ΕΚΤ	ΗΜ/ΝΙΑ	ΜΗΚ. ΠΛΗΚ/ΣΗΣ	ΑΠΟΣΤΑΣΗ ΠΟΝΤ	ΑΔΡΑΝΗΣ ΧΡ.	ΣΩΣΤΑ	ΛΑΘΟΣ	ΣΥΝΟΛΟ ΣΧΟΛΙΩΝ
User 1	2	15/2/2008	2721	32452	34	0	0	214
User 2	1	15/2/2008	2890	33923	313	0	0	282
User 3	1	15/2/2008	3613	50307	153	0	0	339
User 4	1	15/2/2008	2270	41414	72	0	0	272
User 5	1	15/2/2008	4451	52757	311	0	0	450
User 6	1	15/2/2008	3433	40113	214	0	0	334
User 7	1	15/2/2008	2996	51637	179	0	0	300
User 8	1	15/2/2008	2534	31068	65	0	0	211

Figure 4: Statistical Analysis for "*GATES*" software usability evaluation

ΕΜΦΑΝΙΣΗ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΓΙΑ ΤΟ ΣΕΝΑΡΙΟ Πρόγραμμα Ηλεκτρονικός Lite

ΟΝΟΜΑ ΧΡΗΣΤΗ	No ΕΚΤ	ΗΜ/ΝΙΑ	ΜΗΚ. ΠΛΗΚ/ΣΗΣ	ΑΠΟΣΤΑΣΗ ΠΟΝΤ	ΑΔΡΑΝΗΣ ΧΡ.	ΣΩΣΤΑ	ΛΑΘΟΣ	ΣΥΝΟΛΟ ΣΧΟΛΙΩΝ
User 1	1	15/2/2008	1356	11946	147	0	0	170
User 2	1	15/2/2008	1101	9000	222	0	0	143
User 3	1	15/2/2008	1313	14937	361	0	0	284
User 4	1	15/2/2008	2065	19198	109	0	0	263
User 5	1	15/2/2008	1765	14550	109	0	0	226
User 6	1	15/2/2008	2187	15718	142	0	0	231
User 7	1	15/2/2008	1832	23762	450	0	0	335
User 8	1	15/2/2008	1343	11243	203	0	0	190

Figure 5: Statistical Analysis for "Electronic Lite" software usability evaluation

In these figures, the tables depicted present information concerning the name of the participant, the date that the experiment took place, the total length of the string typed using the keyboard and the mouse buttons, the total distance covered by the mouse cursor, the idle time and finally the number of general comments and comments on wrong or right actions.

5. **Discussion**

In this paper *Usability Logger* v2.3 software tool is presented and its use in a case study concerning the usability evaluation of two software applications discussed. From the analysis, the distinguishing characteristics of the software tool mentioned are relating to the easiness with which the software can be used in different working environments, the recording and calculation of the distance covered by the cursor and finally the form that displays all data collected together with screenshots.

As aforementioned, *Usability Logger* can be employed in various environments where the software application under evaluation is installed. This easiness of use is very convenient for usability evaluation tests since such tests can be performed in usability laboratories, as well as in environments where the software application under evaluation is used. Furthermore, the tool can be used at the same time in more than one workstations employed in a usability evaluation test.

Another feature worth mention is the recording of the cursor movements. This is a feature that is included in most of the software mentioned in Table 1. *Usability Logger* can measure the distance covered by the mouse cursor in specified time slots, thus calculating mouse movement for specific instances and present the total distance covered. This feature is very useful in quantitative analysis and more specific in cases where the exact distance covered by the mouse during particular tasks is required.

Finally, an important characteristic is the presentation of all the data collected in a form (Figure 2). In this form, screenshots from the interaction, for every time interval that we have selected, are presented and for every screenshot, relevant information together with comments from the usability expert is presented. This is a very convenient presentation in the phase of the analysis.

Acknowledgements

We would like to thank Molina Alfaro Alexandra and Anastasiou Stiliano for they valuable contribution to the development of the previous versions of *Usability Logger.* This work was funded by the European Union - European Social Fund (75%), the Greek Government - Ministry of Development - General Secretariat of Research and Technology (25%) and the Private Sector in the frames of the European Competitiveness Programme (Third Community Support Framework - Measure 8.3 - programme PENED - contract no.03EΔ832).

References

[1] Abran, A., Khelifi, A., Surynu, W., Seffah, A., 2003. Usability Meanings and Interpretations in ISO Standards, Software Quality Journal, Vol. 11, pp. 325–338.
[2] Nielsen, J., 1993. Usability Engineering, Academic Press, London.
[3] Dix, A., Finlay, J., Abowd, G., Beal, R., 2004. Human-Computer interaction, Prentice Hall, UK.
[4] ISO 9241-11. Ergonomic requirements for office work with visual display terminals – Guidance on usability, 2003.
[5] ISO/IEC 9126. Software Product Evaluation—Quality Characteristics and Guidelines for the User, 2001.
[6] Usability Logger software tool: http://quality.eap.gr
[7] "Electronic Lite" software: http://3tee-rodou.dod.sch.gr/Electronicos.htt

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-405

Towards Compositional Safety Analysis via Semantic Representation of Component Failure Behaviour

Konstantinos MOKOS [a], Panagiotis KATSAROS [a], Nick BASSILIADES [a], Vangelis VASSILIADIS [b], Maxime PERROTIN [c]
[a] *Department of Informatics, Aristotle University of Thessaloniki, Greece*
e-mails: mokosko@otenet.gr, {katsaros, nbassili}@csd.auth.gr
[b] *Gnomon Informatics S.A., Thessaloniki, Greece – e-mail: v.vassiliadis@gnomon.com.gr*
[c] *European Space Agency, ESTEC/TEC-SWE (Software & Standards), Noordwijk, The Netherlands*
e-mail: Maxime.Perrotin@esa.int

Abstract. In dependable systems engineering safety assessment of complex designs that involve software and hardware components is one of the most difficult tasks required. Due to the different modelling languages and models that are used for complementary tasks, the model and specification artefacts are not easily shared by the experts involved in the design process. Moreover, the structural and semantic differences of the used language representations open a possibility for inconsistencies between the corresponding models. This work explores the role of an ontology representation of component failure behaviour as a basis for automated model transformations, as well as a library of reusable knowledge artefacts to be used in different modelling languages and models. The presented approach was motivated by recent findings and requirements derived from European industrial-driven research and development projects[1].

Keywords. Ontology, Model Driven Software Development, Dependability, Safety Analysis

1. Introduction

In the design of complex critical systems, safety analysis has to be performed in a systematic manner and it is usually refined and updated iteratively as the design process proceeds. Safety assessment traditionally requires the combination of various results derived from various models. Due to the different modelling languages and models that are used for complementary tasks, software engineers research model transformation techniques [10] that enable them to cope with:

- the heterogeneity in textual representation, syntax, semantics, and scope of the modelling languages and
- the possible inconsistencies that are likely when we have concurrent development of models on the basis of different tools.

[1] This work is partly funded by the European Space Agency (ESA) Contract Ref: RFQ3-12080/07/NL/FM/na.

This work explores the role of an ontology representation of component failure behaviour as a basis for automated model transformations, as well as a library of reusable knowledge artefacts to be used in different modelling languages and models. We present the architecture of a client application that will generate *AltaRica specifications* from models described in the *Architecture Analysis & Design Language (AADL)*. AADL is a textual and graphical language introduced by the Society of Automotive Engineers (SAE) for the design and analysis of the software and hardware architecture of safety critical real-time systems. It can be used in conjunction with an *Error Model Annex* standard to add dependability-related information such as fault and repair assumptions and error propagations to an architecture model. AltaRica is a dependability language for formally specifying constraints automata that describe the behaviour of systems when faults occur. Several dependability tools can process AltaRica models, like for example symbolic simulators, model-checkers, fault tree generators, sequence generators and AltaRica graphical modellers (Cecilia OCAS & ARBOR), which are used to support industrial safety analysis processes.

To limit the effort of building dependability models several studies [7, 9] propose the definition of error models that will be applied to a number of system components and will be tailored with component-specific information. This approach assumes the maintenance of libraries with error models and component definitions that will be reused from one project to the other. Most of these projects potentially involve a number of geographically distributed suppliers, subcontractors, developers, certifiers and other stakeholders and this raises the need for an open knowledge base, which will enhance collaboration and information sharing intra and inter organizationally. We invest on an ontology based representation of component failure behaviour that supports the exchange of models among the different modelling tools and the involved stakeholders and opens prospects to meet the following requirements:

- To define a formal semantics in order to capture the meaning of error model definitions in a correct and complete manner.
- To support error model definitions easy to process by tools through an XML-based serialization format (which is not supported in current AADL tools).
- To define a formal semantics that will be deployed as a network resource, in order to validate error model definitions against a central schema (similar to the validation of XML documents against their XML schema).
- To provide a facility for describing the relationships with alternative models and languages in addition to AltaRica.
- To allow queries for potential solutions to design problems on the Web.

In Section 2 we report the findings of related previous work. Section 3 introduces basic concepts of compositional safety analysis and describes the AADL and AltaRica languages with their associated tools. Section 4 outlines the ontology based representation of component failure behaviour. Section 5 provides a simple case study to show the capturing of error models in AADL architecture designs and their transformation to AltaRica specifications. We conclude by summarizing the scope of the ongoing work towards the development of the outlined Semantic Web based safety engineering process.

2. Related work

The ASSERT European Integrated Project (Automated proof-based System and Software Engineering for Real-Time systems) has investigated, elaborated and experimented advanced methods to integrate failure propagation Altarica models to system designs developed in AADL.

During this project, a major concern was raised related to the possibility of inconsistencies between the used failure propagation models, Altarica models and the other models that were developed according to the ASSERT process. In order to avoid these inconsistencies, Onera investigated the feasibility of a tool that generates an Altarica model from a model described in AADL and proposed the development of libraries of Altarica nodes that will be reused from one project to the other. The tool was based on a model transformation approach that extracts from the AADL model, the functional and hardware architecture of the system. The development of libraries worked well only for families of systems that did not differ too much, and generally, it was found difficult to model certain types of failure propagations in physical domain.

Similar to Onera, LAAS investigated, during the first phase of the ASSERT project, an alternative approach where an AltaRica specification is enriched with failure propagations coming from AADL code written following the Error Annex standard. According to the ASSERT reports the Altarica code was found to be much more complex than before, but feasible, as long as component relationships and various kinds of analysis are defined [9].

3. Compositional Safety Analysis

A service delivered by a system is its behaviour as it is perceived by its users and in the safety analysis domain this is well known as the external state. Correct service is delivered when the service implements the intended system function. Related to the abnormal system operations, three basic concepts are used: failure, error, and fault. A system fails when it deviates from the intended functions and behaviours. Service failure means that at least one or more external states of the system deviates from the correct service state. The forms of deviations are characterized in terms of failure modes. An error is the part of total system states that can result in a system failure. Error events are used to model internal state (behaviour of basic component). A failure occurs when an error propagates from internal states to external states, causing the whole system to fail. The causes of an error, either adjudged or hypothesized, are referred to as faults. A fault produces one or multiple errors when it is activated. A fault is active when it causes an error, otherwise it is considered as dormant [1, 2, 4].

Failure condition (FC) refers to a combination of failure modes applied to functions of the system under study. We considered the following generic failure modes of functions: total loss, partial loss and erroneous behaviour. The FC may also include conditions that describe the current mode of the system. A FC may be permanent or transient [9] and it is characterized by the following attributes:

- SEVERITY classification that includes the categories Catastrophic, Hazardous, Major, Minor and No Safety Effect.

- QUANTITATIVE OBJECTIVE, which is a failure rate value that can be stated per mission hour, or for a given mission phase. Typical values are 10^{-9} /hour for Catastrophic FC, 10^{-7}/hour for Hazardous. Other categories are dependent on project, reliability objectives, etc.
- QUALITATIVE OBJECTIVE essentially describes the number N of individual faults which are considered for a given FC: “No combination of events with less than N individual faults shall lead to FC” with N = 2 for Catastrophic FC, N=1 for Hazardous and Major FC.

Safety analysis provides information about the consequences of components/ system failures and is related to all other types of analysis. The aim is to identify hazards, to access the risk, and to support hazard control. The analysis requires error models and models capturing failures and environmental conditions. Classical techniques for safety analysis are the so-called HAZOP (Hazard and Operability Studies), FMEA (Failure Modes and Effects) and FTA (Fault Tree Analysis) [4, 5].

3.1. AADL

The SAE Architecture Analysis & Design Language (AADL) is a textual and graphical language used to design and analyze the software and hardware architecture of safety critical real-time systems [3, 4]. AADL is used to describe functional interfaces to components (such as data inputs and outputs), performance-critical aspects (such as timing), how components interact, how application software components are allocated to execution platform components, as well as the dynamic behaviour of the runtime architecture (by supporting the concept of operational modes and mode transitions).

AADL is designed to be extensible. By allowing users to extend the core language with additional features it is also possible to specify safety levels, criticalities, fault tolerance, and error handling. Extensions take the form of new properties and analysis specific notations that can be associated with components. The user just imports Annex libraries which extend the language and customize an AADL specification to meet project- or domain-specific requirements. AADL is currently provided with two basic annex libraries: the behaviour annex and the error model annex [7].

The Error Model Annex defines additional properties that describe reliability of the system components and a state machine that specifies error states. Thus the Error Model Annex can be used in conjunction with the description capabilities of AADL to add dependability-related information, such as fault and repair assumptions, error propagations and fault-tolerance policies. More precisely, error model types are used for modelling fault/error/failure behaviours. They may declare error states (possibly identified during a hazard analysis), failure modes (possibly identified during a Failure Mode and Effects Analysis), error events (used to model internal faults), as well as internal repairs and error propagations (which are used to model failure effects).

The Error Model Annex supports a compositional approach to error modelling:

- It enables reuse of error models.
- Makes it easier to modify architecture specifications and automatically regenerate safety and reliability models.

- It facilitates abstraction and mixed-fidelity modelling.
- It enables improved traceability between architecture specifications and models and analysis results.

The language features can be used to specify errors that propagate between components and connections, depending on the structure of the architecture model. There also language features used to specify how components vote to detect and mitigate errors in their subcomponents or in the components on which they depend. Provided the availability of appropriate tool support the language semantics can be used to check for consistency, completeness and traceability between the error models of interacting components, as well as between the error models of components and their subcomponents.

AADL is supported by two open source tools: OSATE and TOPCASED. The latter is a Metamodelling Framework developed by Airbus and 20 companies and currently supports AADL Graphics, AADL XML, model transformation and the Behaviour Annex. The former is an Eclipse based tool developed by the Software Engineering Institute (SEI) and currently supports full language editing and semantic checking, multiple analysis plug-ins and integrated text and graphical editing with TOPCASED.

3.2. AltaRica

AltaRica [6] is a dependability language designed in the University of Bordeaux to formally specify constraints automata (formal models which describe the behaviour of systems when faults occur). Several dependability tools can process AltaRica models [4, 5, 6, 9]: symbolic simulators, model-checkers, fault tree generators, sequence generators, AltaRica graphical modelers (Cecilia OCAS & ARBOR), etc. In AltaRica the models have the following structure:

- A model of a system consists of hierarchies of components called nodes. A node gathers flows, states, events, transitions and assertions.
- A component has is own variables (internal or external), plus some flow variables it can only read and are shared with other components. Flows are the visible parameters of the component whereas states are internal variables.
- Assertions are boolean formulae that state the constraints linking flows and internal states. Assertions are system invariants (they must always be true).
- Transitions describe how the initial states may evolve. They are characterized by a guard (a boolean constraint over the component flows and state), an event name and a command part (value assignation of some state variables).
- The system is primarily separated to physical subsystems, which are described as nodes and capture all events and behaviour of the subsystems.

Our goal is the fault model to have as much information as possible about the various events in which the components of the system can malfunction. A failure can be defined as an event which can affect the state of the node. Failure modes are called the different forms of deviation from correct service and can be defined using a transition taken to the particular failure event. An AltaRica model-checker checks the set of properties that eliminate the failure events. Safety requirements are given as

failure conditions specified in the form of Linear Temporal Logic formulae that have to be proved. An AltaRica fault-tree generator takes an AltaRica model as input and an unexpected event and subsequently generates a fault-tree describing the failure situations that lead to the unexpected event. Once the fault tree is generated, a fault tree analyzer can be used to compute the set of prime implicants on the non-temporal failure conditions.

4. Ontology based representation of component failure behaviour

The presented modelling languages (AADL and AltaRica) are both specification languages with different textual representations, syntax and scope, thus creating a wide semantic gap. Most current model transformation approaches (e.g. Model-driven software development [10] and OMG's Model-driven Architecture) address the problem of different model representation formats by the use of metamodels, which in fact can express only the logical syntactical structures of the corresponding models and cannot specify their semantics [12]. However, true model integration can only be achieved on a semantical level [8].

Ontologies are considered a key element for semantic interoperability because they act as shared vocabularies that utilize a grammar for using this vocabulary. The grammar specifies (with formal constraints) what it means to be a well-formed specification and how the terms in the ontology's controlled vocabulary can be used together. The semantics of the concepts of interest is specified in a logic-based representation language thus resulting in unambiguous and machine-processable specifications. Moreover, the used representation languages come with executable calculi enabling querying and reasoning at run time.

The proposed safety engineering process (Figure 1) uses a domain ontology built in the Web Ontology Language (OWL) to provide a classification of all possible component failure modes [15] and associate them with constraints that state their meaning. The whole architecture includes two bindings that specify the semantic mapping from the AADL/Error Annex and AltaRica metamodels to the domain ontology. An AADL/Error Annex model is an instantiation of the corresponding metamodel and it is semantically interpreted by the underlying domain ontology. The logical and syntactical structures of the AADL models should obey to a number of metamodel constraints. These constraints are associated with OWL property types and the semantic mappings to the domain ontology are used by reasoners that apply inference rules on the relationships of the ontology concepts. Inference rules will help to detect lacks of model elements and semantically inconsistent parts. Rules will also infer additional assertions, which are not specified explicitly but will be derived through the AADL and failure mode semantics (e.g. derived error models [7]). We use the term model inference to refer to the aforementioned model validation, which is possible only in an ontology-based model transformation process.

Reasoning functionality will be exploited also in semantic web clients other than the AADL to Altarica model transformation. One obvious prospect is the definition of the notion of component failure behaviour matching based on relationships like refinement and simulation [13].

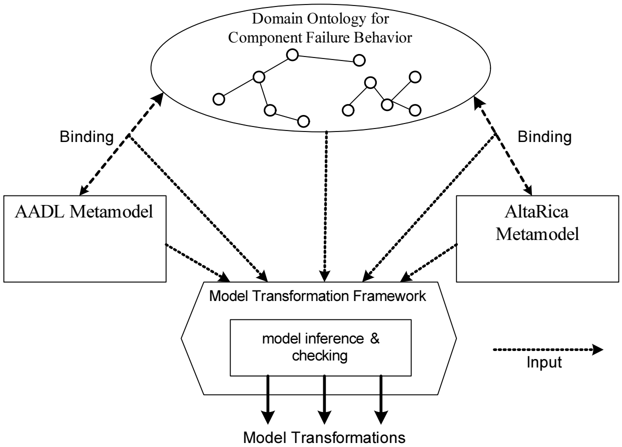


Fig. 1. Model transformation architecture

In the proposed framework, additional rules related to the syntactic mapping of AADL models to AltaRica specifications will be implemented within the used model transformation tool chain (e.g. openArchitectureWare [11]). Interoperability between the model transformation tool chain and the ontology part of the shown architecture will be achieved through the EMF Ontology Definition Metamodel [14] that essentially provides a set of programming APIs for creating, modifying, and navigating an OWL ontology in which all objects are also treated as Eclipse Modelling Facility (EMF) model objects. This makes possible to utilize the comprehensive development facility of Eclipse that is required by the selected model transformation tools.

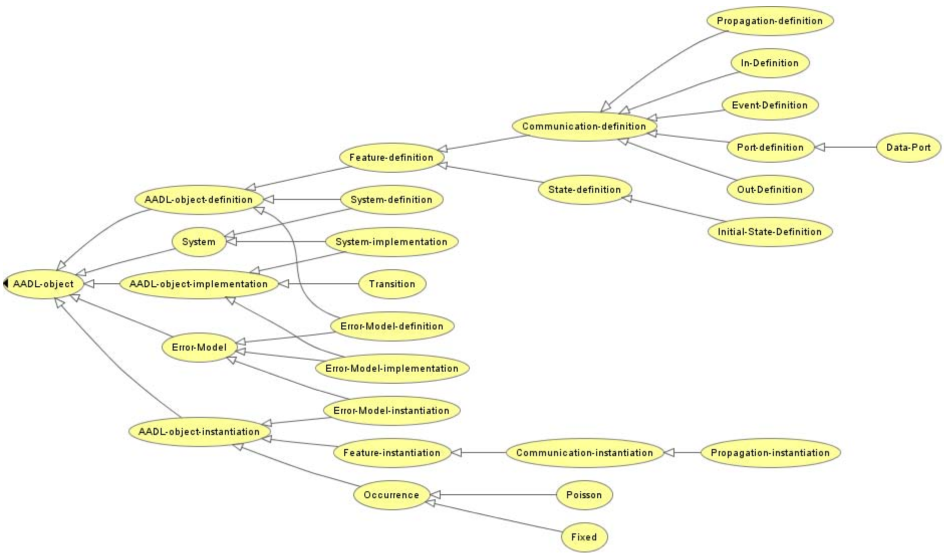


Fig. 2. Class hierarchy of the AADL/Error Annex ontology

Figure 2 shows the class hierarchy of the AADL/Error Annex ontology and Figure 3 presents a snapshot of the instances in the model transformation case study of the next section. The ontology follows a meta-modelling approach, defining classes for

artefact definitions, implementations and instantiations. Actually two major artefacts are modelled, namely system and error model. Systems can consist of nested sub-components. System and error model definitions have features, namely in/out ports, error events-propagations and states. Definitions are associated with implementations, which can further refine the models. Error model implementations define transitions between states, based on incoming events. Finally, artefact instantiations associate artefact implementations with a specific pattern of event/data flow occurrence.

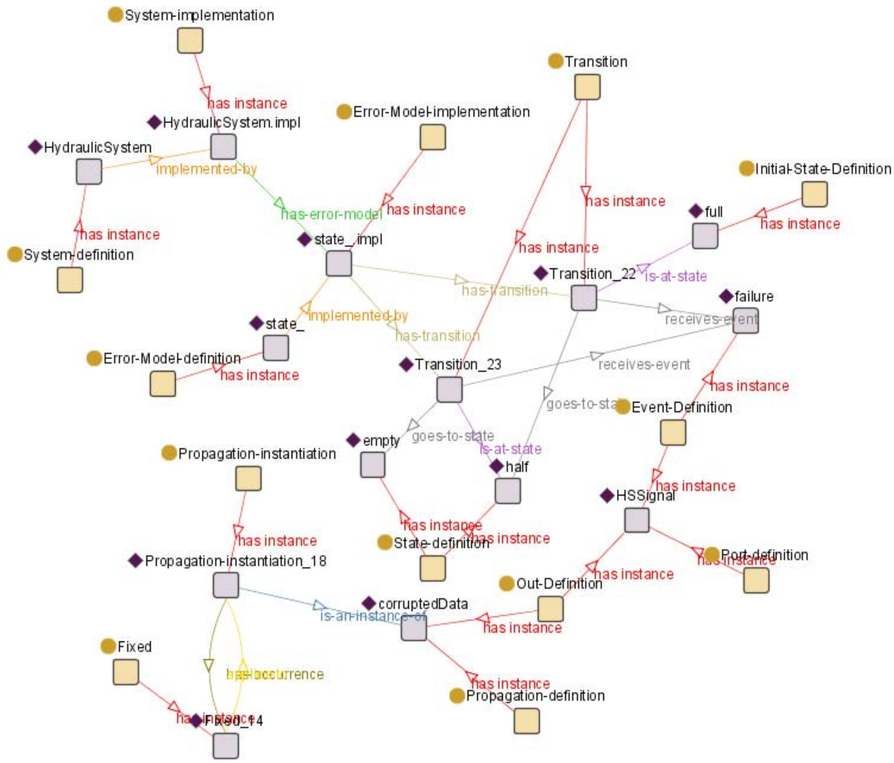


Fig. 3. Ontology representation of an Hydraulic System failure behavior

5. Case study

Our prototype architecture will be tested by simple and more complex case studies. A quite simple system design is the Aircraft Hydraulic Leakage Detection System shown in Figure 4. The purpose of this system is to detect and stop leakages in the two hydraulic systems, which feed the moving parts of an aircraft. Leakages in the hydraulic systems could in the worst case result in such low pressure that the airplane becomes uncontrollable. To avoid this, some of the branching oil pipes are protected by shut-off valves. These valves can be used to isolate a branch in which the leakage has been detected. Then, although the leaking branch will no longer function, the other branches will still keep the pressure and be able to supply the moving parts with power.

The reading of oil level sensors and the controlling of the four shut-off valves are handled by three electronic components, as depicted in Figure 4. The H-ECU (Hydraulic Engine Control unit) is a software component that continually reads the oil

reservoir levels of the two hydraulic systems (HS1 and HS2) and determines which shut-off valve to close accordingly. HS1 sensors determine valves B1 and C1, while HS2 sensors determine valves B2 and C2. However, it would be very dangerous if some electrical fault caused more than one valve to close at the same time. For this reason, two programmable logic devices, here called PLD1 and PLD2, continually read the signals to and the statuses of the valves, and if the readings indicate closing of more than one valve, they will disallow further close operations. Thus, PLD1 and PLD2 add fault tolerance to the shut-off subsystem implemented in the H-ECU. PLD2 will only accept a request from the H-ECU for closing a particular valve if the check, which is partially done in PLD1, indicates that everything is fine. A valve will close only when both the low-level signal, which is the shut-off signal directly from the H-ECU, and the high-side signal, which is the checked signal from PLD2 are present.

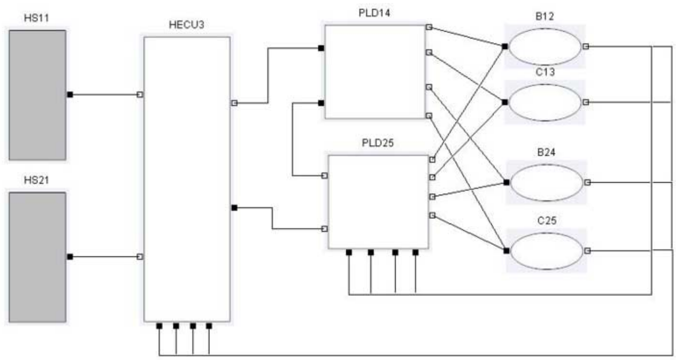


Fig. 4. Hydraulic leakage detection subsystem

The Hydraulic Leakage Detection System was modelled in AADL using the OSATE (Eclipse) graphical interface and the model was subsequently transformed to an AltaRica specification. The specification excerpt of Figure 5 represents the Hydraulic System (HS).

Figure 3 shows the AADL/Error Annex ontology representation of the Hydraulic System component. There are two definition objects, one for the system `HydraulicSystem` and one for the error model `state_`, and two equivalent implementation objects. These are connected to the required feature, transition and state objects. Finally, there is one instantiation object for the occurrence pattern of the corrupted data error propagation.

6. Conclusion

This paper introduces a Semantic Web architecture that supports geographically distributed industry groups in performing compositional safety analyses on components/systems during the requirements definition and design phases. We presented the details of the model transformation architecture under development, but we also stated the scope of our proposal that in fact is versatile:

- To support product/component manufacturers in sharing specifications of component failure behaviours intra and inter-organizationally and in looking for components that match a particular failure behaviour.

- To cope with the heterogeneity in syntax, semantics, and scope of the modelling languages used in compositional safety analyses and the possible inconsistencies, that arise when we have concurrent development of models with different tools.

<pre>system HydraulicSystem
  features
    HSSignal: out event port;
  annex error {**
        error model state_
        features
        full: initial error state;
        half,empty: error state;
        failure: error event;
        CorruptedData: out error propagation
{Occurence => fixed 0.1};
        end state_;
 	
  **};
end HydraulicSystem;
 	
system implementation HydraulicSystem.impl
  annex error {**
error model implementation state_.impl
transitions
        full- [failure] -> half;
        half- [failure] -> empty;
end state_.impl;
        **};
end HydraulicSystem.impl;</pre>
	<pre>node HDL_HS1
	 flow
	  icone : [1,1] : out ;
	  HSSignal : HDL_type_HS : out ;
	 state
	  state_ : {full,half,empty} ;
	 event
	  failure ;
	 init
	   state_ := full ;
	trans
	state_ = full |- failure -> state_ := half;
	state_ = half |- failure -> state_ := empty;
	assert
	(if state_ = full then HSSignal = Full
	 else if state_ = half then HSSignal = Half
	else HSSignal=Empty);
	extern
	   law <event failure> = exponential(1e-1) ;
	edon</pre>

Fig. 5. AADL and AltaRica specifications for the HS1 and HS2 component failure behavior

7. References

[1] Leveson, N. G. Safeware: System safety and computers. Addison-Wesley Publishing Company, 1995

[2] Avizienis, J.-C., Laprie, Landwehr, B. Basic concepts and taxonomy of dependable and secure computing, *IEEE Transactions on Dependable and secure Computing*, Vol. 1 (1), pp. 11-33, 2004

[3] Society of Automotive Engineers, SAE Architecture Analysis and Design Language, SAE standard AS5506. Available in: http://www.aadl.info.

[4] Joshi, A., Miller, S. P., Whalen, M., Heimdahl, M. P. E. A proposal for model-based safety analysis, 24th Digital Avionics Systems Conference (DASC), 2005

[5] Bieber, P., Castel, C., Seguin, C. Combination of Fault Tree Analysis and Model Checking for safety assessment of complex system, 4th Europ. Dependable Computing Conf., LNCS 2845, pp. 19-31, 2002

[6] Safety assessment with AltaRica – Lessons learnt Based on Two Aircraft System Studies. 18th IFIP World Computer Congress, Toulouse, France, 2004

[7] Feiler, P., Rugina, A. Dependability Modeling with the Architecture Analysis & Design Language (AADL), Software Engineering Institute (SEI), 2007

[8] Dietrich, J., Elgar, C. An Ontology Based Representation of Software Design Patterns, In: Design Pattern Formalization Techniques, Toufik Taibi (Ed.), IGI Publishing, pp. 258-279, 2007

[9] Bieber, P., Blanquart, JP., Durrieu, G., Lesens, D., Lucotte, J., Tardy, F., Turin, M., Seguin, C., Conquet, E. Integration of formal fault analysis in ASSERT: Case studies and lessons learnt, 4th European Congress on Embedded Real Time Software (ERTS 2008), Toulouse, France, Jan 2008

[10] Stahl, T. and Volter, M. Model-driven software development: Technology, Engineering, Management, Wiley, 2005

[11] The Official openArchitectureWare Homepage, http://www.openarchitectureware.org/

[12] Saeki, M. and Kaiya, H. On Relationships among Models, Meta Models and Ontologies, Proceedings of the 6th OOPSLA Workshop on Domain-Specific Modeling (DSM'06), Un. of Jyväskylä, Finland 2006

[13] Pahl, C. An ontology for software component matching, International Journal of Software Tools Technology Transfer, Vol. 9, Springer, pp. 169-178, 2007

[14] EODM - EMF Ontology Definition Metamodel, http://www.eclipse.org/modeling/mdt/?project=eodm

[15] McDermid, J. and Pumfrey, D. A development of Hazard Analysis to aid software design, 9th Annual Conference on Computer Assurance (COMPASS), 1994

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-415

Impact of Comment Statement on Code Stability in Open Source Development

Hirohisa AMAN [1], Hirokazu OKAZAKI
Department of Computer Science, Ehime University, Japan

Abstract. Comment statements written in a source code are important information to understand and to maintain the source code. This paper performs an empirical study of impacts of comment statements on source code stability in an open source development. The study collected $4,521$ source code and their version-upgrade data from a major open source software project, Eclipse. The empirical results show the following findings: (1) a source code with a high comment-density (about more than 40%) has a tendency to be stable through the upgrdes, but (2) a source code with frequently-written comments has the opposite tendency. This work thus concludes that describing many comments is recommend for developing a stable code, but frequently describing comments is discouraged.

Keywords. Comment statement, Code stability, Metrics, Empirical approach

1. Introduction

Code review is an effective means for developing a high-quality software[1]. However performing careful code review for all software components is difficult because of some practical constraints such as a shortage of technical experts and a lack of time to review. Those constraints pose a potential risk for releasing a defective software. To reduce the risk, many software vendors/organizations have recommended and/or obligated their software engineers to write helpful comments in their source code. In general, comment statements are useful to enhance the understandability of source code[2,3], and writing adequate comments promotes potential code review by their programmers themselves. Therefore a source code with adequate comments is expected to be reliable and to be stable during the maintenance stage of the software. However most empirical work on software maintainability in the literature has used design metrics and/or code (executable code) metrics, and comment statements have been erased before the measurement or ignored throughout the analysis. The contribution of this paper is to present an empirical study concerning an impact of comment statement on a source code maintenance, especially a code stability in an open source software development. Moreover, this study has another aim related to comment description. Although it is expected that comment

[1] Corresponding Author: Hirohisa AMAN, Dept. of Computer Sc., Ehime Univ., 3, Bunkyo-cho, Matsuyama, 790-8577 Japan; E-mail: aman@cs.ehime-u.ac.jp.

statements are desirable in a quality code, there is another remark that many comments in a code would be a “code smell”[4] for refactoring the code, since that code would be difficult to understand without those comments. To explore such effects of comment statements on the code quality, our empirical study will focus on not only the amount of comments written in a source file but also frequency of comment descriptions.

2. Comment Statement Metrics

In our empirical study, we focus on two different aspects of comment statements in a source code—comment density and frequency. We now define the following metrics for representing comment density and frequency.

2.1. Comment Density

First, we introduce two metrics for representing comment density.

Definition 1 (COM_L)
Define “COMment density in Line count” (COM_L) of a source code as follows:

$$\mathrm{COM}_L = \frac{\text{number of lines including a comment statement}}{\text{total line count}} \ ,$$

where the following exceptional comment statements are omitted:

- comment out of code,
- copyright description, and
- application-specific configuration, e.g., “$NON-NLS” in Eclipse[5].

□

Definition 2 (COM_C)
Define “COMment density in Character count” (COM_C) of a source code as follows:

$$\mathrm{COM}_C = \frac{\text{number of characters contained in comment statements}}{\text{total character count}} \ ,$$

where white-space characters are counted in; the exceptional comment statements are omitted as in COM_L. □

The more comments are written in a source code, the higher values in COM_L[2] and in COM_C are observed.

2.2. Comment Frequency

We next define a metric for representing a frequency of comment description in a code.

Definition 3 (COM_F)
In a source code, the statements are categorized into either (1) “comments” state or (2) “others” (usually, executable code) state. Now consider the state transitions (1) → (2) and

[2] COM_L can be measured using an existing tool such as RSM[6].

```
BufferedReader r = ... ;  // reader
int x = 1;  // line number
String a = null;  // one-line content
while ( (a = r.readLine()) != null ){
    // print line number and content
    System.out.println(x + " : " + a);
    x++; // incrementing line number
}
```

Figure 1. Code fragment sample (a).

```
BufferedReader reader = ... ;
// print line content with line number
int number = 1;
String line = null;
while ( (line = reader.readLine()) != null ){
    System.out.println(number + " : " + line);
    number++;
}
```

Figure 2. Code fragment sample (b).

(2) $\rightarrow$ (1) by going through from the head to tail of the source code. Define "COMment description Frequency" (COM_F) of a source code as follows:

$$\mathrm{COM}_F = \frac{\text{number of the state transitions between "comments" and "others"}}{\text{total line count}} . \qquad \square$$

The transition count represents a frequency of comment description. The more frequently describing comments throughout a code, the higher value in COM_F is observed. For example, Figs.1 and 2 show the same code, but their comment descriptions are different: COM_F of (a) $= 10/8$, and that of (b) $= 2/8$. A frequent comment description is represented by the number of transitions between code and comments: for example in Fig.1, "**int** x = 1;" $\rightarrow$ "// line number" $\rightarrow$ "String a = null;" ...; To compare code in different sizes, we divided the number of transitions by the total line count.

Although many comments may be desirable in a quality code, Fowler[4] has pointed out describing many comments is a "code smell" for refactoring the code. Figure 1 shows a simple example of code who contains many meaningless comments; the comments for variables could be remove by replacing their names to appropriate ones, and it would be sufficient to write the single aim of fragment. To examine such remarks,we perform an empirical study using the above metrics in the following section.

3. Empirical Study

3.1. Impact on Change Rate

We used a major open source software, Eclipse[5], as our experimental object. For the upgrades of versions "2.0 $\rightarrow$ 2.1," "3.0 $\rightarrow$ 3.1" and "3.1 $\rightarrow$ 3.2," we collected $4,521$ source code and their version-upgrade data, where our code are limited to middle or large scale[3] ones, and our upgrade data are limited to ones by corrective maintenance activities. We measured the before-upgrade code with COM_L, COM_C and COM_F, and calculated the change-rate δ $(= (\text{changed LOC})/\text{LOC})$ for each upgrade.

Figure 3 shows an impact of the comment density COM_L to the change-rate δ, where the moving average of δ is drawn for the code corresponding to that $\mathrm{COM}_L \pm 0.05$— for example, the moving average of δ of code whose COM_L values are in $[0.35, 0.45]$ is drawn for $\mathrm{COM}_L = 0.4$ in the figure. We found a trend which δ decreased with

[3]The code whose LOC ≥ 67 are considered to be a middle of large scale ones, where the threshold value 67 is determined empirically; that is the median of LOC.

increasing COM_L. A Wilcoxon test[4][7] showed the decreasing tendency is statistically significant. A similar tendency was identified about COM_C as well[5].

We next focused on only source code whose comment density is relatively-high such that $\mathrm{COM}_L > 0.4$, and analyzed an impact of the comment description frequency COM_F to the change-rate δ. Figure 4 shows the moving average of δ for the code corresponding to that $\mathrm{COM}_F \pm 0.025$. We found a trend which δ increased slowly with increasing COM_F. The slowly increasing tendency is statistically significant with a Wilcoxon test[6].

From the above results, we conclude as follows: a source code having a high comment-density can be upgraded with a small change, i.e., such code is to be stable. However a code in which comments are "frequently" described may have an opposite tendency on the code stability.

3.2. Impact on Change-Proneness

We also explored an impact of comment statements on change-proneness for Eclipse source code. Let us consider the following two sets of code:

- Set A: set of code that has never been changed from Ver.2.0 to Ver.3.2;
- Set B: set of code that was changed soon after the release.

We performed a Wilcoxon test on the difference of comment-density COM_L between the set A and the set B. The test showed that there exists a significant difference in COM_L between A and B; COM_L of code in set A is grater than that of set B, with the significance level of 5%. A similar result was identified for COM_C as well. Although we also performed a Wilcoxon test for COM_F, we could not obtain any significant difference between A and B. The reason would be that δ-change was slowly as we saw in the above section.

Therefore we identified a tendency that a code having a high comment-density is less change-prone, i.e., more stable through the version-upgrade.

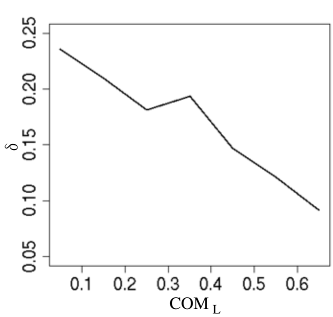


Figure 3. Change-rate δ versus COM_L.

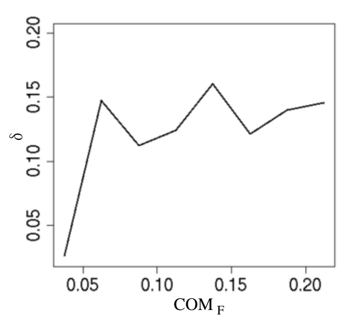


Figure 4. Change-rate δ versus COM_F.

[4] All source code are categorized into two sets: set of $\mathrm{COM}_L < 0.4$, and set of the others. The test examined whether there exists a significant difference between those sets or not, with the significance level of 5%.

[5] The details are omitted for the lack of space.

[6] We used only two sets of source code: set of code whose COM_F is in the lower 25%, and set of ones whose COM_F is in the upper 25%, since chnges in δ were "slowly". The test examined whether there exists a significant difference between those sets or not, with the significance level of 5%.

4. Conclusion and Future Work

This paper has performed an empirical study of impacts of comment statements on code stability in a major open source software, Eclipse. The $4,521$ source code and their version-upgrade data were collected from the development site. The statistical analyses using the three comment-description metrics showed the following findings: (1) a source code having a high comment-density (about more than 40%) is likely to be stable through the software upgrades; (2) a code in which comments are frequently described has a tendency to be less stable even if the code has a high comment-density.
Therefore, many comments are recommended to make a stable code, but a frequently describing comments is discouraged in terms of the code stability.

Our future work includes an impact analysis of blank lines on code stability, and a further study on code change patterns such as refactoring. Moreover, explorations of comment consents and of object class structure will be important future work:

1. Comment contents
   We focused on comment description statistics, i.e., density and frequency, in our empirical study. However comment contents and their semantics will also play important roles in the code maintenance. We will need to take into account comment contents as well. A morphological analysis and/or data mining of comment contents would be helpful means in our future work.
2. Object class structure
   We did not consider class structure in our empirical work. Usually a method definition has a corresponding documentation comment (Javadoc), and a class including many small methods would have a high-level comment frequency. Such class may be change-prone because of many methods. We will need to explore not only statistics of comment frequency but also each code contents (class structures) or class design metrics such as the number of methods, and perform further analyses on the relationship of code change rate with comment frequency.

Acknowledgment

The authors would like to thank the anonymous reviewers for their thoughtful comments.This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 2007-2008, 19700028.

References

[1] G. Sabaliauskaite, S. Kusumoto and K. Inoue, Extended metrics to evaluate cost effectiveness of software inspection, *IEICE Trans. Inf. & Syst.* **E87-D** (2) (2004), 475–480.
[2] S.L. Pfleeger, *Software Engineering: Theory and Practice*, Prentice-Hall, London, 2001.
[3] H. Aman, H. Okazaki and H. Yamada, An effect of comment statements on source code corrective maintenance, E. Tyugu and T. Yamaguchi (Eds.), *Knowledge-Based Software Engineering*, IOS Press, Amsterdam (2006), 135–138.
[4] M. Fowler, *Refactoring: Improving The Design of Existing Code*, Peason Education, NJ, 1999.
[5] Eclipse, http://www.eclipse.org/.
[6] RSM Metrics, http://msquaredtechnologies.com/m2rsm/docs/rsm_metrics.htm .
[7] S. Peter, *Quick Statistics: An Introduction to Non-Parametric Methods*, New York, 1981.

This page intentionally left blank

Decision Support Methods for Software Engineering

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-423

ARM: Actor Relationship Matrix

Komon Ibe[1], Shinobu Saito[2], Shuichiro Yamamoto[1]
[1]*Research Institute for System Science, Research and Development Head Quarters, NTTDATA, Tokyo, JAPAN*
[2]*Center for Applied Software Engineering, Research and Development Head Quarters, NTTDATA, Tokyo, JAPAN*

Abstract. The SD model which the i* framework includes is used to analyze and describe the requirements among the stakeholders in the business. However, the practical method which supports the stakeholders for checking whether or not the requirements in the SD model are comprehensive is not proposed. This paper introduces a method which we called the ARM (Actor Relationship Matrix). The ARM is two-dimensional matrix that defines the relationship among actors. We also discuss the effectiveness of the proposed method which contributes comprehensive review of requirements.

1. Introduction

The i* framework [1] is said to be one of the best-known goal oriented requirements analysis [2]. The SD (strategic dependency) model in the i* framework expresses the requirements dependency among the actors in business. However, the practical method, which could help the stakeholders check whether or not the requirements in SD model are comprehensive, is not proposed.
This paper proposes a method to define and analyze requirements from the view point of the relationships among actors by using a two-dimensional matrix, which is we called the ARM (Actor Relationship Matrix). And the effectiveness of this method is also discussed.
In section 2, the sample of SD model and its issues are introduced. This issues occur when the i* framework is applied to actual problems. Section 3 presents the ARM which we aim to adopt to address the issues. It is discussed that the ARM delivers three types of reviews to address the issues in section 4. Section 5 introduces the related work. And then, the summary and future work is describes in section 6.

2. SD model and its issues

A product sales management service is used as an example of the SD model. Requirements descriptions of this service are followings.

R1: The customer shall place an order to the system for a product.
R2: The system shall check with the warehouse staff to see if the product is in stock.
R3: The ordering staff shall instruct the system to make up for low inventory.

R4: The system shall instruct the warehouse staff to ship the product from warehouse.
R5: The warehouse staff shall confirm that the product is shipped.
R6: The warehouse staff shall update condition of product inventory.
R7: The delivery staff shall deliver an invoice to the customer.
R8: The delivery staff shall deliver a shipment-list to the customer.
R9: The customer shall pay a fee to the cashier.

R1-R4 and R7-R9 describe relationships between actors. Fig 1 illustrates the SD model with based on these requirements description. R5 and R6 describe actor's own goal. So these two requirements are not shown in the SD model.

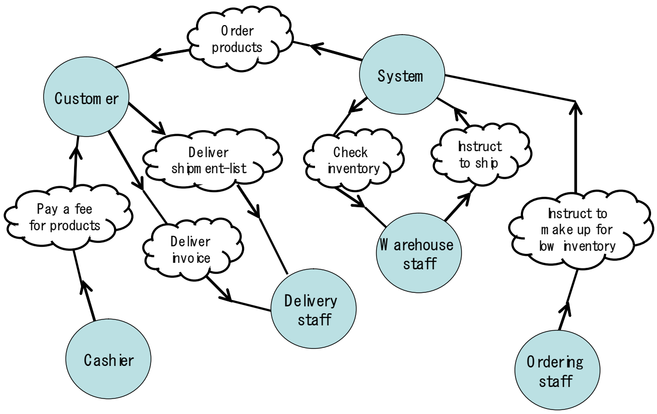


Fig 1 SD model

When the i* framework is applied to actual problems, the number of actors to consider is very large and then the relationships among actors are complex. For that reason, it is difficult for stakeholders in the business to check efficiently whether or not the requirements have been comprehensively described in SD model.

3. ARM (Actor Relationship Matrix)

The ARM (Actor Relation Matrix) is two-dimensional matrix that defines the relationship among actors. The above requirements descriptions are shown in table 1 using the ARM. The element at Row i, Column j of the table ($i \neq j$) describes the requirement what the beneficiary side (actor i) expects of the provider side (actor j). For example, in the earlier SD model, the system has the requirement "Check inventory" of the warehouse staff. So this requirement is written in the space in table where the system's row and the warehouse staff's column intersect.
In diagonal space (those where i=j), the actor's own goal to achieve is written. This applies to R5 and R6 in requirements descriptions. So the warehouse staff has two goals, "confirm shipment status" and "update product inventory" are entered in the space corresponding to the row and column of the warehouse staff. The ARM could be used to organize the requirements among all actors in the business. So ARM makes it possible to perform a comprehensive review of requirements.

Table 1 ARM

Demander \ Supplier	Customer	System	Warehouse staff	Ordering staff	Delivery staff	Cashier
Customer					"Deliver shipment-list" "Deliver invoice"	
System	"Order products"		"Check inventory"			
Warehouse staff		"instruct to ship"	"Confirm shipment status" "Update product inventory"			
Ordering Staff		"Instruct to make up for low inventory"				
Delivery staff						
Cashier	"Pay a fee for products"					

4. Early experience

ARM was applied to the requirements review for a certain financial system specification to evaluate the effectiveness of ARM. The specification included 14 Actors and 152 requirements. The 3 developers studied ARM in one hour and reviewed specifications in 2 hours. The number of Actors was same before and after the review. 71 Requirements were added and 33 requirements were deleted. After review, there were 190 requirements in total.

The participated system developers evaluated ARM as the effective method to check the completeness of requirements specifications. Moreover, they said that ARM was easy to use.

5. Discussion

The ARM supports stakeholders to review the requirements descriptions. Using sample of the ARM (table 1), three types of reviews which the ARM could provide are below.
(Type 1) Review of requirements expected of actor
Nothing is written in ARM's the cashier column. This means that the other actors do not have any requirements of the cashier. The ARM shows that after the cashier has collected a fee from the customer, there is no requirements description of how the casher manages that fee. As a response to this case, a new requirement by the system could be added, namely, "Confirm the deposit of money" of the cashier. The new requirement is written in the space where the system row and cashier column intersect.
(Type 2) Review of requirements expected by actor
Nothing is written in the ARM's the delivery staff row. This means that the delivery staff does not have any requirements of the other actors. In this service the delivery staff shall deliver shipment-list and invoice. Ordinarily, such an action would happen because another actor has instructed the delivery staff to deliver. In other words, the

delivery staff needs instruction that notices the timing of the delivery. However, the current requirements descriptions are missing these instructions. To react this case, a new requirement expected by the delivery staff could be added, namely “Instruct to deliver the product” of the warehouse staff. The new requirement is written in the space where the delivery staff row and the warehouse staff column intersect.

(Type 3) Review of the actor’s own goals

Nothing is written in the diagonal element spaces of the ARM for any actor except the warehouse staff. This means that the goal to be achieved by the actor’s activities is missing. It is necessary to reconsider the significance of the actor itself, that is, why the actor is necessary.

6. Related work

The NFR framework [3] proposes some catalogues based on quality requirements standpoint (i.e. Security, Efficiency) to support system design. But in the NFR approach, actors who are related to the business are not considered. The reduction method of redundant requirements (soft-goal) is already proposed by [4]. And Eric et al. [5] introduce an approach using both i* framework and BMM [6] for intentional model and analysis. However, comprehensive review method of requirements derived from actors is not proposed yet.

7. Conclusion

In this paper the ARM which is supporting for review of requirements descriptions is proposed. ARM defines relationships among actors in the business and addresses the issues of application of i* framework. Then it is also proposed that three type reviews is delivered using the ARM. This contributes to comprehensive review of requirements descriptions.

The ARM would be easy for anyone to implement because it uses a familiar two dimensional matrix. So it is a powerful technique for beginner. From now on, we will investigate practical implication from case studies in order to demonstrate the benefits of ARM.

References

[1] i* : http://www.cs.toronto.edu/km/istar/
[2] Refsq2007, Trondheim, Norway, June 11-12, 2007, Proceedings
[3] Brian A. Nixon, Eric Yu, John Mylopoulos, Lawrence Chung, “Non-Functional Requirements in Software Engineering,” Kluwer Academic Publishers (Springer),1999
[4] Shuichiro Yamamoto, Goal Algebra for IT Alignment, proc. of JCKBSE 2006, pp.147-150
[5] E. Yu, M. Strohmaier, X. Deng, Exploring Intentional Modeling and Analysis for Enterprise Architecture, Proceedings of the Workshop on Trends in Enterprise Architecture Research (TEAR'06), 2006.
[6] Business Motivation Model: http://www.omg.org/docs/dtc/06-07-01.pdf

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-427

Impact coverage analysis on requirement changes

Shinobu Saito
Research and Development Head Quarters, NTTDATA, Tokyo, JAPAN

Abstract. In system development project, when a single requirement in the cascade model of requirements is changed, many other requirements may be changed. Although requirements management tools are used, it is not clear how to ascertain the potential change requirements at the time when a requirement changes. This paper proposes a method to define the proper impact coverage based on requirement dependency. And the proposed method also defines the proper change sequence of requirements. That may support for reducing rework in change management.

1. Introduction

In information system development project, various requirements are defined by requirements engineers. One high level requirement is linked to several low level requirements. In other words, several low level requirements are depend on one high level requirement. That is, these requirements are described as cascade model. So, when a single requirement is changed, many other requirements may be changed. The impact of the single requirement change may be distributed not only downward but also upward requirements in cascade model. Therefore, change management is the most complex issue on requirements engineering [1]. For a proper change management, it is necessary for requirements engineers to define the method to ascertain the impact coverage in response to the requirements dependency in the cascade model, when a single requirement change is raised.
In this paper, section 2 defines the requirement algebra to express the relationship between the higher level requirements and the lower level requirements. Next, the impact coverage analysis using the requirement algebras is proposed in Section 3. And then, discussion and Issue of the proposed method are considered.

2. Requirement algebra

When the requirement x is high level requirement of the requirement y, namely x is parent requirement of y, the requirement algebra is defined as following.

$x \rightarrow y$

Both right and left set of requirement algebra can have more than one requirement. For example, when the requirement x is high level of both requirement y and z, requirement algebra of this relationship is following.

$x \rightarrow y + z$

Using above requirement algebras, the cascade model in Figure 1 is transformed to list

of the requirement algebra. In this paper, this requirement algebra list is called as the RDL (Requirements Dependency List). The RDL of Figure 1 is following.

RDL= {a → b + c, b → d + e, c → e + f}

We define that the requirement not linked to high level requirements is called as the root requirement. A requirement algebra list has only one root requirement.

In Figure 1, Requirement a is root requirement

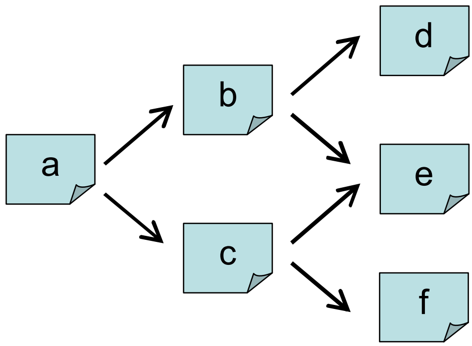


Figure 1 Example of Cascade model of requirements

3. Impact coverage analysis

3.1. Downward impact coverage

When a single requirement x changes, downward impact is occurred in cascade model. That is, all lower level requirements that flow down from requirement x are potential change requirements. In this paper, these down ward potential change requirements list is called as DIL (Downward Impact List). The root requirements of This DIL is requirement x.

In Figure 1, when requirement b changes, the DIL of this case is following.

DIL = {b → d + e}

3.2. Upward impact coverage

Besides downward impact, upward impact is also occurred in cascade model. When a single requirement x changes. This change may make requirements engineers to renegotiate at higher level requirements from requirement x. If a higher level requirement y changes, all lower level requirements that flow down from requirement y also become to be potential change requirements. In this paper, these upward potential change requirements list is called as UIL (Upward Impact List). The root requirements of This UIL is requirement y. For avoiding overlap of algebras, UIL does not include the requirement algebras which are already listed in the DIL.

In Figure 1's case, if the change of requirement b causes the change of requirement a, the UIL of this case is following.

UIL = {a → c, c → e + f}

3.3. Integration of downward and upward impact coverage

In this paper, all potential change requirements list is called as WIS (Whole Impact List). WIS is created by integration of DIL and UIL. For proper integration, new requirement algebra is added. On developing the UIL the requirement x caused the change of requirement y. Therefore, we define this new relationship as new requirement algebra. This algebra is called as BA (Bridge Algebra). BA is following.

BA = {x → y}

And then, WIL is organized as follows.

WIL = DIL + BA + UIL

In Figure 1, the BA and the WIL of this case are following.

BA = {b → a}

WIL = DIL + BA + UIL

= {b → d + e} + {b → a} + {a → c, c → e + f}

= {b → a + d + e, a → c, c → e + f}

With reference to above WIL, we can describe the impact coverage model on the change of requirement b in Figure 2. This model shows the change sequence of the requirements with based on requirement dependency of the cascade model (Figure 1).

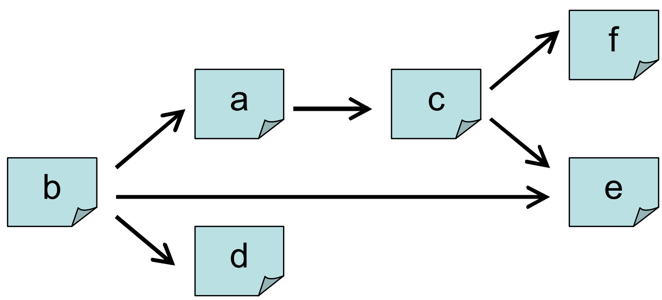


Figure 2 Impact coverage model on the change of requirement b in Figure 1

4. Discussion

4.1. Proper change Sequence

Utilizing the impact coverage model, requirements engineers can know both the proper sequence and the dependency relation to reconsider potential change requirements.

In Figure 2, after the change of requirement b, the proper change sequence of requirements is following.

Step1. Both requirement a and requirement d are reconsidered depending on the change of requirement b.

Step2. Requirement c is reconsidered depending on the change of requirement a.

Step3. Requirement f is reconsidered depending on the change of requirement c. And requirement e is reconsidered depending on the change of both requirement c and requirement b.

If requirements engineers do not reconsider in above sequence, the rework of reconsidering the potential change requirements may happen.
For example, if the requirement f is changed before the change of requirement c is completed, over time requirement f may be reconsider once more after the change of requirement c is completed.

4.2. Merging requirement

When a requirement is listed in both DIL and UIL, this requirement is affected by both downward impact and upward impact. In this paper, this requirement is called as a merging requirement. In Figure 2, requirement e is merging requirement. At the time when requirements engineers consider the change of requirements, they should take care of the timing of completion of change at the merging requirement point in impact coverage model. If not, rework of reconsidering the potential change requirements may happen.
For example, if the requirement e is changed before the change of requirement c is completed, over time requirement e may be reconsider once more after the change of requirement c is completed.

5. Conclusion

This paper proposed the method to ascertain the impact coverage at the time when a requirement changes in the cascade model of requirements. The proposed method is based on requirement dependency in the cascade model. The impact coverage model is transformed from the cascade model by the application of the proposed method. And the impact coverage model shows the proper impact coverage and the proper change sequence of requirements. This sequence supports for reducing rework in requirement changes. This paper shows the cascade model which is composed of requirements linked to different abstraction level requirements. Besides this model, the proposed method also could be applicable to cascade model which has relationship between different abstraction levels of requirements. Namely, if the requirement b and requirement c in figure 1 are linked (e.g. b → c), we could get the proper change sequence by adopting the proposed method.
Although commercial requirements management tools [2] [3] show the relationship based on requirement dependency in cascade model, any function to teach the proper change sequence is not proposed yet. So far, requirements engineers have to decide the change sequence of requirement on their own way, when requirement is changed. The proposed method will support to streamline requirements engineer's work.
It is also necessary to examine the effectiveness of the method for the practical information system development.

References

［1］ Elizabeth Hull, et al., Requirements Engineering second edition, Springer, 1994.
［2］ Telelogic, Doors: http://www.telelogic.com/products/doors/index.cfm
［3］ IBM, RequisitePro: http://www-306.ibm.com/software/awdtools/reqpro/

Knowledge Management for Business Processes, Workflows and Enterprise Modeling

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-433

A Design Method for the Enterprise Knowledge Management Architecture

Masakazu Kanbe and Shuichiro Yamamoto
Research Institute for System Science, NTT DATA CORPORATION

Abstract. We propose a method for designing an enterprise knowledge management architecture. Our method includes the processes to develop KM assumptions, survey KM Task, evaluate KM assumptions, select KM tasks to be improved and develop KM road map. This method uses the hierarchical knowledge sharing architecture consists of business vision, knowledge process and knowledge management environment to execute knowledge management activities. We illustrate the effectiveness and limitation of our method.

Keywords. Knowledge management, knowledge process design, corporate knowledge sharing

1. Introduction

Knowledge creative company is attracted as the service sector of economy is growing. Although knowledge management methods were applied, it was not sufficient to share and integrate knowledge among different divisions of each company. Reasons of the problems are as follows. First, knowledge sharing process is not sufficiently considered whereas the knowledge management concentrates only for extracting explicit knowledge from tacit knowledge. The First problem is studied by many researchers widely in the contexts of community of practice and interest. Our approach includes the consideration for community of practice or interest as some parts of knowledge management process. Second, individual knowledge management activities are isolated and relationship among them is not managed. This paper mainly discusses the first problem. The second problem is the future work. In this paper we try to propose a design method for the enterprise knowledge sharing architecture.

The proposed method evaluates knowledge management assumptions based on the facts, and then extracts the important tasks for realize the assumptions in the hierarchical knowledge sharing architecture. The hierarchical knowledge sharing architecture consists of business vision, knowledge process and knowledge sharing environment. Finally it helps design hierarchical road map of tasks for the enterprise knowledge sharing.

In section 2, we propose the design method for the enterprise knowledge sharing architecture. Section 3 provides an example of the design method for a company. Discussions are studied in section 4. Related works is presented in section 5.Summary and future issues are described in Section 6.

2. Enterprise knowledge management architecture design method

The process of the enterprise knowledge sharing architecture design method is shown in figure 1. The process includes the following 5 steps.

(STEP1) KM assumptions are developed. The assumptions will be used to prioritize the knowledge management tasks based on the observations of the effectiveness of these tasks.

(STEP2) The effectiveness of KM tasks is surveyed using questionnaire and interview.

(STEP3) Evaluate KM assumptions with evidences derived by the facts gathered from the above surveys.

(STEP4) Select the KM tasks that have low score in the evaluation. In this step we use the knowledge management architecture that has three layers, i.e., knowledge management strategy layer, knowledge management process and environment layer. The knowledge management environment layer includes business rules and IT tools that support knowledge management processes.

(STEP5) Develop KM Road Map based on tasks extracted before. In this step, knowledge management tasks are interrelated in the 3 layer architecture.

These knowledge management processes are repeated iteratively.

1.Develop KM Assumptions

2. Survey KM Tasks

3.Evaluate KM Assumptions

4.Select KM Tasks to be improved

5.Develop KM Road Map

Fig.1 Enterprise Knowledge Management Design Method

For large companies, we may have many knowledge management tasks and observation facts for them. In these large knowledge management cases, we should have a method to reduce the numbers of facts and tasks for promoting knowledge management. To resolve the problem, we develop assumptions for describing the knowledge management architecture to be. The assumptions are then evaluated and assigned with the weights that are calculated from the related facts. If an assumption "a" is related to the fact "f", the weight W (a, f) is defined as follows.

W (a, f) = 2 if "a" is supported by "f"
= 1 if "a" is partially supported by "f"
= -1 if "a" is not supported by "f"
= 0 otherwise

The total weight of the assumption "a" is defined as follows.

TW (a) = Σ W (a, f) where "f" is a fact

We also define W (a, L) as Σ W (a, f) where "f" is a fact of knowledge management layer L. As we have 3 knowledge management layers, the following expression holds.

We now can decide the order of assumptions by the value of TW (a). A threshold value V is used to extract the weak assumptions WA = {a| TW (a) <V}.

Then we can extract candidate generic tasks based on the relationship between assumptions and specific tasks.

ST (a) = {t| an assumption "a" is supported by task "t"}

WST = {t| for an "a" in WA, "a" is supported by task "t"}

GT (a) = {g| "g" is the generic task for a specific task of ST (a)}

CGT = {g | "g" is an element of GT (a) where "a" is an element of WA}

WGT = {g | "g" is the generic task for a specific task of WST}

Obviously, WGT =CGT.

The road map will be developed using the generic tasks defined by CGT to solve the problems elicited by the facts.

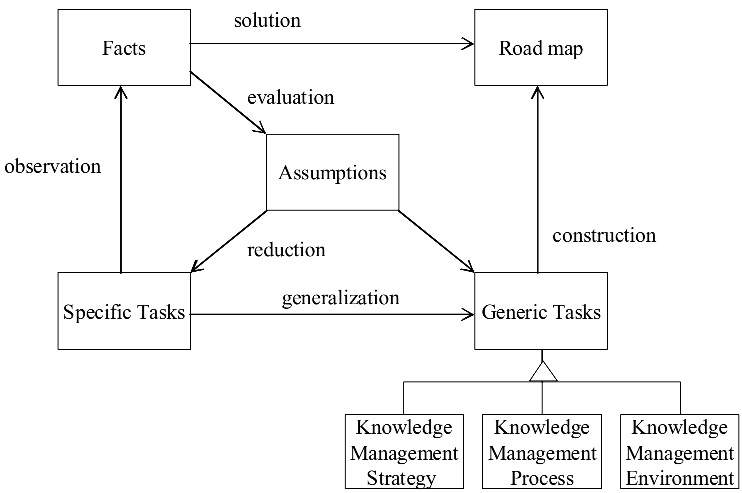


Fig.2 Relationship of knowledge management constituents

Figure 2 shows the relationship of the above knowledge management design constituents. The current specific knowledge management tasks are observed and facts on these tasks are gathered in survey reports. The assumptions are developed to extract insufficiently supported tasks by evaluating facts as described before. After selecting specific tasks to be improved, these tasks are transformed to generic tasks for describing knowledge management road map. The transformed generic tasks are categorized hierarchically into strategic, process and environment levels. For developing this hierarchy, we can use generic task patterns such as Knowledge Networks proposed by Enkel et al [1]. Table1 shows the hierarchical generic tasks based on the Knowledge Networks.

(1) Knowledge management strategy layer

In this layer, the following strategic tasks are defined. Enterprise knowledge vision provides a knowledge management goals and a road map to achieve them. The road

map provides the inter-relationship among knowledge management tasks. Involvement of employees and Commitment of top management are necessary tasks for the success of enterprise knowledge management. Establishing knowledge management board is also important for the successive knowledge management. The board manages total enterprise knowledge management program. Environmental factors of knowledge management should be also supported by the board and top management if necessary.

(2) Knowledge management process layer

This layer has two kinds of knowledge management processes. The first type is knowledge sharing works. Facilitation of knowledge sharing work helps individual employee share knowledge. Development of knowledge worker collaboration helps employee break the wall that exists between different organizations. Organization of knowledge collaboration processes changes the legacy work style for enabling effective knowledge sharing. Integration and coordination with external organization enables to develop knowledge sharing network with other organizations.

The second type processes are necessary for maintaining enterprise knowledge management processes. These processes include identification of knowledge management success factors, investigation of knowledge management performance, and selection of knowledge management tasks that the improvement is necessary. These second type processes are used for designing the enterprise knowledge management architecture.

(3) Knowledge management environment layer

This layer includes two things for knowledge management. The first is organization policies and guidelines. And the second is Information and communication support tools. By using this table showing candidates for knowledge management generic tasks, we can easily describe knowledge management road map in an appropriate level.

Table1. enterprise knowledge management architecture based on the knowledge networks by Enkel et al[1].

layer	No	Generic task for knowledge management
Knowledge management strategy	1.1	Enterprise knowledge vision
	1.2	Establish knowledge management board
	1.3	Supportive environmental factors
	1.4	Involvement of employees
	1.5	Commitment of top management
Knowledge management process	2.1	Facilitation of knowledge sharing work
	2.2	Development of knowledge worker collaboration
	2.3	Organization of knowledge collaboration processes
	2.4	Integration and coordination with external organization
	2.5	Identification of knowledge management success factors
	2.6	Investigation of knowledge management performance
	2.7	Selection of tasks to be improved
Knowledge management environment	3.1	Organization policies and guidelines
	3.2	Information and communication support tools

3. Example

In this section, we will explain the design process of an enterprise knowledge management road map using the proposed method. The study is based on the case of a Japanese large-scale information service company in Tokyo.

3.1 Target knowledge management activities

The enterprise had 30 knowledge management tasks. The IT management division of the company wanted to evaluate and integrate current activities. They also try to develop the road map based on the new vision of knowledge management. As they had already many and different current knowledge management activities, it was difficult to develop it.

And they also had 5 survey reports shown in Table 2. These reports included 100 facts on the current knowledge management activities. The top management interview had 10 facts and related to knowledge management strategy. The knowledge management and employee satisfaction questionnaire had 78 facts for knowledge management processes. The enterprise SNS (Social Networking Services) and enterprise portal questionnaire had 12 facts for knowledge management environment tools.

These facts are useful to evaluate the effectiveness of the current knowledge management activities as well as extract problems of the current knowledge management status.

Table 2.Survey and extracted facts

surveys	Number of facts			KM layers
Interview with top managements	10	S	4	Strategy 38
		P	5	
		E	1	
Knowledge management questionnaire	57	S	15	Process 38
		P	28	
		E	14	
Employee satisfaction questionnaire	21	S	12	
		P	4	
		E	5	
Enterprise SNS questionnaire	7	S	7	Environment 24
		P	0	
		E	0	
Enterprise portal questionnaire	5	S	0	
		P	1	
		E	4	
Total	100			

3.2 Knowledge management assumptions

In the first step of the proposed method, we will develop the knowledge management assumptions. The assumptions are useful to verify the effectiveness of the current activities. We stated the following 5 assumptions on the case study for the evaluation.

A1: The intranet information is completely able to use.

A2: The level of information sharing can be managed flexibly.
A3: The relationship between knowledge user and content can be managed.
A4: Specialists are mutually collaborated.
A5: From the viewpoints of user, knowledge management tasks are well integrated.

The intention of A1 is that business goals are solved by effective use of intranet information. The A2 describes that the appropriate knowledge expression is necessary for different knowledge. For example, the method of sharing explicit knowledge for document is different from those of tacit knowledge. The A3 says that the knowledge classification should be suitable for business processes. The A4 shows that promotion of a corporate culture that employee collaborate each other is effective. The A5 says that integration and usability of knowledge management activities are effective for business management. It is necessary to develop the knowledge management vision and road map to realize the vision. These assumptions are useful to develop the knowledge management vision.

3.3 Evaluate knowledge management tasks

(1)Gathering facts for evaluating KM assumptions

In the case study we had 5 surveys and 100 facts as shown in Table 2 and 3.1. The table shows also the relationship between surveys and knowledge management layers. There were 38, 38, and 24 facts respectively for the strategy, process, and environmental layers. The top management interview includes 4, 5, and 1 fact respectively for the tree layers. Because the enterprise SNS was just introduced, the SNS questionnaire included only the facts for the strategy layer. On the other hands, the enterprise portal questionnaire included no strategy layer facts. This is because the portal has long history for the enterprise knowledge management in the company.

Table 3. Evaluation of assumptions

	A1	A2	A3	A4	A5	Total
W(*. Strategy)	-2	5	-3	0	-3	-3
W(*. Process)	-6	-9	-14	-9	-4	-42
W(*. Environment)	-3	4	-1	4	-7	-3
TW(*)	-11	0	-18	-5	-14	-48

(2) Evaluation of the assumptions

Table 3 shows the evaluation result of assumptions by the facts. The assumptions are A1, A2, A3, A4, and A5.

If we set threshold V = -10, then WA = {A1, A3, A5}. The assumptions A1, A3, and A5 are necessary to be improved.

The table also shows the knowledge management process layer has many problems, because the total score of the layer is the worst one. Many enterprises have same problem for the legacy knowledge management programs. The problem is as follows. Although many knowledge management services were introduced, all company and systematic effectiveness is not satisfied because there is no collaboration process.

3.4 Select knowledge management tasks to be improved

On the left hand side, table 4 shows the relation between 30 tasks and selected assumptions, A1, A3, and A5. On the right hand side, it also shows the relation between 30 tasks and the generic knowledge management tasks in table 2.

First, ST (A1), ST (A3), and ST (A5) are extracted and then they are merged. In this way we get the following task set WST to be improved.

WST = {1, 2, 3, 6, 9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 26, 27, 28, 29, 30}}

In this case, we reduced 11 of 30 tasks.

Then we get the following generic task set WGT by using right side of the table.

WGT = {1.1, 1.3, 2.1, 2.2, 2.3, 2.4, 2.6, 3.1, 3.2}

Now we are ready to develop the road map for the enterprise knowledge management. The WGT will be the base of the road map.

Table 4. Task matrix

Task-ID	A1	A3	A5	knowledge managemet strategy	knowledge management process	knowledge managemet environment
1			O		2.3	3.1
2	O	O	O	1.3	2.1	3.2
3	O	O	O		2.1	3.2
4					2.6	
5					2.6	
6	O	O	O	1.1	2.1,2.2	3.2
7					2.6	
8						
9	O	O	O		2.1	3.2
10	O				2.1	
11	O		O		2.1	
12		O			2.2	3.2
13			O		2.4	3.1
14			O		2.4	3.1
15		O	O		2.4	3.2
16	O	O	O	1.1	2.1	3.2
17					2.6	
18	O	O	O		2.1,2.3	3.2
19					2.6	
20		O		1.1	2.6	
21					2.6	
22					2.6	3.1
23					2.6	
24					2.6	
25					2.6	
26	O	O	O			3.2
27		O	O	1.1	2.2	3.1
28	O	O	O			3.2
29	O	O	O		2.2	3.2
30	O	O	O		2.1	3.2

3.5 Develop knowledge management road map

The developed road map is shown in figure 3. The strategy layer has enterprise vision and supportive factor. The knowledge process layer has four generic tasks. These tasks are facilitation of knowledge work, knowledge worker collaboration, organization of knowledge collaboration, and collaboration with the external organization. In this layer, we also add knowledge management process 2.6, although we have only one task 20 for the facts from survey reports. The investigation of knowledge management process is achieved by interview and currently is not supported by tools in the knowledge environment layer. The result is used for enterprise knowledge vision.

The knowledge environment layer contains specific tasks, because the generic task is too broad for understanding the objective tasks. We chose these tasks from WST.

Extracted organizational rules are human resource management, information sharing guideline, and collaborative work style. Information and communication tools are enterprise SNS, enterprise mail magazine, and enterprise portal services.

In this way, it is easy to realize the enterprise knowledge management steps. The relationship between tasks indicates the necessary condition. By using the relationship, the validness of the road map can be logically explained.

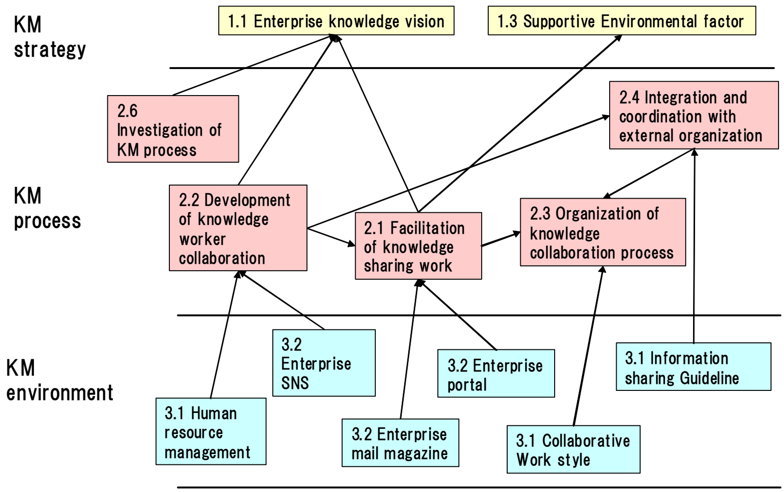


Fig.3 Knowledge management road map

4. Discussions

4.1 Effectiveness

The characteristic of the proposed method is that it can select and concentrate the important knowledge management tasks based on the evaluation of the current knowledge management activities. In fact, it reduced 33.6% of tasks.

The design method also provides a way to use many surveys form different points of view. This is possible by defying assumptions on knowledge management. The assumptions are used to evaluate the realities of the enterprise knowledge management. Our method is based on the facts obtained from company knowledge management surveys. Company knowledge management surveys reflect company business goal, needs and opportunities. In the case study 100 facts from 5 survey reports were used to evaluate 5 assumptions.

The method also inter-relates knowledge management tasks in logical and hierarchical knowledge management architecture. In this way, the case study showed that the design method could develop knowledge management road map efficiently.

4.2 Limitation

The proposed method has the following limitations, although it has useful characteristics. These are future issues of the proposed enterprise knowledge management design method.

The design process of the knowledge management assumptions is not described. It is necessary to elicit the assumptions from enterprise business goals.

The evaluation of facts on each assumption is now subjectively-based. Towards the objective evaluation, some guidelines are necessary to describe practical evaluation procedures.

Currently, tasks are extracted if one assumption is supported by them. More strict extraction methods can be defined. For example, the following set WSTM select lesser tasks than those of WST.

WSTM = {t| majority of WA support the task “t”}

We need further research to clarify the best task selection methods.

The goodness of the road map on the enterprise knowledge management is also necessary to evaluate in the real knowledge management operation.

In this paper, we do not mention about the knowledge development. The enterprise knowledge is separated and not integrated among different knowledge management tools. Integration of enterprise knowledge is necessary for enterprise knowledge management, although we proposed an integrated knowledge management process design method. It is desirable to combine our process design method with existing knowledge development methods.

We have the enterprise SNS, mail magazine, and portal in figure 3. It is useful to combine these tools and search the enterprise knowledge in seamlessly. We therefore need tool integration method for enterprise knowledge management.

5. Related works

The knowledge network [1] proposed by Enkel et al. pointed the usefulness of the hierarchical architecture of knowledge management activities. However, they did not concern the inter relationship among activities. Moreover, their method is specific for the knowledge management for the case of M&A of companies. Although we applied their hierarchical approach to our design method, we revised it suitable for the enterprise knowledge management architecture. Because the knowledge network was very conceptual, it did not contain the road map as well as the design process based on the evaluation of current knowledge management activities.

Nonaka et al. [2] proposed the middle up down knowledge management process. The management style can be a candidate of generic collaboration process task 2.4 in table 2.

Kokune et al. [3] proposed the Fact Based Collaboration Modeling integrating Balanced Score Card and Goal Analysis. Although they also use the hierarchical layers include financial management, business process and IT services, they do not consider the knowledge management activities. However, their concept of fact based modeling can be used for extracting valuable facts for knowledge management activities.

The community of practice [4] provides useful approach to exchange knowledge in the self directed way. Although the approach is not hierarchically structured, this can also be considered as an important knowledge sharing task in our approach. Our method will be used to enhance the community of practice for company knowledge management.

Dieng et al. [5] surveyed methods for corporate knowledge management. They showed that corporate memory management methods have 6 tasks, i.e., detect needs, build memory, distribute it, use it, evaluate it and make evolve. However, they did not

concern the hierarchical architecture of these tasks. And they did not consider the integration of different knowledge management tasks in the company.

Kulkarni et al. [6] proposed the knowledge management maturity model. However they did not consider the knowledge management design process.

Tiwana and Ramesh [7] proposed the methodology for developing knowledge management system. Their methodology is effective in defining the relationship between tool components and organization processes to be supported. They suggested intranet tools to support organization processes, such as, distribution, connectivity and publishing." Their approach can be synergistically integrated with our method.

6. Conclusion

In this paper, the enterprise knowledge management design method is proposed. It also shows the usefulness of the 3 layer knowledge management architecture by the realistic large knowledge management case. The paper exemplified the effectiveness of the design method by the case study. The study showed the design method could develop 13 interrelated knowledge management tasks from 30 existing specific tasks using 100 facts and 5 assumptions. Although the knowledge management road map was designed well, it is necessary to evaluate it in the process of executing the road map for the company. It is also necessary to apply the method for other cases of enterprise knowledge management. Moreover, for improving the quality of knowledge management process, the mutual understanding is necessary among top management, employees, and enterprise knowledge designers.

References

[1] E. Enkel, A. Back and G. von Krogh, The concept of Knowledge Networks for Growth, in A. Back, E. Enkel and G. Von Krough (Eds). "Knowledge Networks for Business Growth," Springer, 2007.

[2] I.Nonaka and H.Takeuchi, "The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation", Oxford University Press, 1995.

[3] A. Kokune, M. Mizuno, K. Kadoya, and S. Yamamoto, "FBCM: Strategy modeling method for the validation of software requirements," Journal of Systems and Software, Vol.80, pp. 314-327, 2007.

[4] E. Wenger, R. McDemott, and W. Snyder, "Cultivating Community of Practice", Harvard Business Scholl Press, 2002.

[5] R. Dieng, O. Corby, A. Giboin M. Ribiere, "Methods and tools for corporate knowledge management", International Journal of Human-Computers Studies 51 (1999),567-598.

[6] U. Kulkarni, , and Freeze, R., "Development and Validation of a Knowledge Management Capability Assessment Model", Proceedings of the 25th International Conference on Information Systems, Dec., pp.657-670, 2004.

[7] A. Tiwana and B. Ramesh, "Integrating Knowledge on Web", IEEE Internet Computing, Vol.5 No.3, pp32-39, 2001.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-443

Leveraging use cases in developing service-oriented systems

Shingo Takada
Dept. Information and Computer Science, Keio University, Japan

Abstract. Service-oriented systems are systems that combine services that are dispersed over the Internet. System development is normally done by defining a business process or workflow which specifies which services are to be invoked in what order. The specification of the workflow can be done in a top down manner, i.e., the developer first develops an initial specification (workflow), searches for Web services, and then refines the workflow so that the Web services can be invoked. In this paper, we propose supporting the development of the workflow (specifically BPEL document) by suggesting previous development results that could be reused. We focus on requirements (specifically use cases) and correlate use case steps with BPEL activities. This information is stored in a repository. When developing a new service-oriented system, the developer first describes a new use case. This is used to search for similar use cases and their corresponding BPEL documents. The developer uses the information to define the new service-oriented system.

Keywords. Service-oriented systems, requirements, software development support

1. Introduction

Service-oriented systems (SoS) are systems that combine services that are dispersed over the Internet. When developing such a system, the developer needs to find services that can be used. Thus much work has been done in the area of service discovery beginning with UDDI [1] to using semantic information [2][3] to using QoS information [4]. But these work are applicable for the latter stages of SoS development, i.e., they can be used once the specification of the workflow has been decided on. Unfortunately, even if the workflow has been specified, it often will need to be modified once the Web services that are to be invoked are decided on. This is because the Web services will rarely completely match what the workflow specifies, i.e., there may be differences in parameters, type, etc. So, we propose to take this process one step back; instead of searching for Web services directly, we search for workflows that may be reused.

Previously, we have done work on software reuse based on use cases [5], where we focused on the fact that class diagrams and code are developed to realize use cases. By keeping traces between use cases, class diagrams, and code, we were able to reuse class diagrams and code through searching for use cases that were similar to what we wanted to develop.

This paper has the same philosophy, i.e., by retrieving similar use cases, we can reuse other documents/code to develop new systems. The difference with our previous work is that SoS use services which are in effect black-box, while our previous work

assumed that the developer can modify code for reuse. Since services are normally provided by a third party, it can only be used *as is*. Furthermore, although our previous work considered the class diagram as the main design model, SoS are often based on workflows.

The rest of this paper is composed as follows. Section 2 gives a brief overview of SoS. Section 3 then describes use case. Section 4 describes our approach and tool. Section 5 discusses our tool, and section 6 makes concluding remarks.

2. Service-oriented systems

Service-oriented systems are based on the collaboration of one or more services. Each service takes input, does some sort of processing and returns an output. The processing may consist of repository access or the calling of other services. Because SoS is a collection of services where the result of a service is often (part of) an input to another service, they are often designed using workflows.

WS-BPEL 2.0 [6] (or just BPEL) is a widely-used XML-based standard for describing SoS. It combines services using various control constructs, with the resulting combination itself being exposed as a Web service.

A BPEL document consists of various parts such as partner links, variables, correlation sets, etc. A major part is the activities part, where various Web services are combined by describing activities. Activities are classified into two types: basic activities and structured activities. Basic activities consist of *invoke*, *receive*, *reply*, *wait*, etc. Structured activities include *sequence*, *if*, *repeatUntil*, etc. These activities are put together to form the workflow, a step-by-step description of what is to be done.

3. Requirements and use case

Requirements is an important part of developing any new system. If the requirements are not known, the developer will not know what to build, or erroneously build a wrong system. There are many ways to describe requirements, including natural language, use cases, and formal descriptions. This paper considers use cases.

Although a use case diagram is well known as one of the diagrams in UML [7], the amount of information that is described there (at least according to the UML specification) is limited. However, this does not mean that developers cannot include more information within each use case. Indeed, proposed details of use case contents are similar [8] [9].

Somé [9] proposed the following fields: *Title*, *System under Design*, *Primary Actor*, *Participants*, *Goal*, *Precondition*, *Postcondition*, *Steps*, *Extensions*. Among these, *Steps* denote the procedure for carrying out the use case. Each step in *Steps* is a tuple [Use-Case-Operation, Extension-Point, Alternatives]. An extension point is a label for inserting a different use case (called extension use case). Alternatives are a set of possible steps after the current step is finished. Use case operation is the main component, and takes the form "[delay-specification] [condition-statement] [determinant] entity operation-reference". The delay specification specifies the amount of time to be delayed before executing the step, and uses keywords such as *BEFORE* and *AFTER*. The condition statement specifies the condition that must hold for the rest of this step to be executed, and takes the form "IF condition THEN …". Determinant is

either *a*, *an*, or *the*. These three (delay-specification, condition-statement, and determinant) are optional. The entity consists of one or more words, and must correspond to an actor (Primary Actor or a Participant), System under Design, or an attribute of an actor or System under Design. Conceptually, the entity acts as the subject of the sentence in the use case step. An operation reference is basically the predicate of the sentence, and consists of at least a verb and may also include a verb object. So, for example, "IF PIN is valid THEN System displays welcome" is a valid step.

In the rest of this paper, for simplification, when we say use case step, we will refer to use case operation, and not the tuple.

4. Reusing BPEL Documents by Searching for Similar Use Cases

4.1. Overview

Use case steps and BPEL document's activities part are, at the core, both composed of steps. There have been work where the description of use cases includes workflow [10] and use cases were derived from business process models [11]. Although in the end how much use case steps and BPEL activities actually have affinity with each other is an open question, BPEL activities do provide a way to realize the use case steps. We focus on this to support using use cases to find BPEL documents that may be reusable.

The overall procedure in our approach is as follows:

1. The developer describes a use case.
2. The developer searches for similar uses cases that were specified for previously developed systems.
3. The developer checks retrieved use cases and corresponding BPEL documents. They are reused if found suitable for the current system, and development continues.
4. After development finishes, the use case(s) and corresponding BPEL document(s) for the newly developed systems is stored in a repository for future use.

We have developed a prototype tool called SUCABS (Support for Use Case and Bpel Search) that supports steps 2 and 4 of the above procedure. In the rest of this section, we first describe how a use case and corresponding BPEL document is stored, focusing on how the correlation between them are obtained. We then describe the search support.

4.2. Storage phase: Correlating use case steps and BPEL activities

Before storing the use case and corresponding BPEL document, they need to be analyzed to determine which steps in the use case corresponds to which activities in the BPEL document. This requires that the use case and BPEL document are described in a standard way. Fortunately, BPEL itself is a standard, so this should not be a problem. This is not the case for use case, as there is no standard format for it. We chose the format specified by Somé, and assume that use cases are described with UCEd [9].

SUCABS takes the specified use case document and BPEL document, and analyzes them to obtain the corresponding BPEL activity for each use case step, and to assign a "keyword" to each use case step as follows:

1. Obtain the values for System under Design, Primary Actor, and Participants by scanning the use case document. These values will be used later in step 5.
2. Find the part of the use case document that contains the use case steps (specifically the *procedure* node).
3. Normally, the first step of a use case is a trigger to start the use case, and the BPEL process normally starts with a receive activity. Consider this as the first correlation.
4. Parse each use case step to obtain the values for the use case step template: "[delay-specification] [condition-statement] [determinant] entity operation-reference".
5. For each use case step, do the following:
   5.1. If there is a delay-specification, first consider only that part. Check to see if there is a corresponding BPEL activity. For example, if the delay-specification takes the form "*AFTER* some-time", correlate it with the BPEL activity *wait* if it exists. Assign the symbol "*AFTER*" as the keyword. For the rest of the use case step, recursively apply step 5.
   5.2. If there is a condition-statement ("*IF* condition *THEN*"), first consider only that part. If the next BPEL activity is *if*, then correlate them. If that is not the case, and if the previous BPEL activity is *invoke*, check if the node defining the links between activities (specifically the *sources* node) contains a transition condition. If a transition condition exists, then correlate the condition-statement to it. Add the symbol "*IF*" as the keyword. For the rest of the use case step, recursively apply step 5.
   5.3. If the entity is the System under Design (or an attribute of it), and the next BPEL activity is *invoke*, correlate them. Add "*SUD* operation-reference" as the keyword, where *SUD* is a symbol and operation-reference is the value that was obtained in step 4.
   5.4. If the entity is the Primary Actor or a Participant, and the previous use case step was correlated with *invoke*, add this use case to the previous one, i.e., consider this as part of an external service invocation. Add "*PA* (or *PART*) operation-reference" as the keyword, where *PA (PART)* is a symbol and operation-reference is the value that was obtained in step 4.
6. Save the use case document, BPEL document, the correlation information, and the keywords in the repository.

Note that in step 5, the determinant (*a*, *an*, or *the*) is skipped. The current implementation of SUCABS does not consider determinants.

Next, note that a use case step is not always correlated with a BPEL activity one-to-one. In some cases (such as when a delay-specification exists), one use case step may correspond to multiple BPEL activities. In other cases (such as when the entity is a Primary Actor or a Participant), multiple use case steps may correlate with one BPEL activity.

When generating the correlations, keywords are generated at the same time. These are to be used during the search phase described in the next section. Since the developer is building a new system, it will be a rare case that the actual actors, e.g., ATM and User, for the new system use case and a previous system use case will match. Thus, instead of keeping the original entity, we use the symbols *SUD*, *PA*, and *PART*.

As for the operation-reference, we keep it as is, and check for synonyms of the verb in the operation-reference during the search phase.

4.3. Search phase: Searching for similar use cases

Our approach is based on searching for similar use cases. Since each use case is associated with a BPEL document, this means that the corresponding BPEL document will also be available. Furthermore, the correlations between use case steps and BPEL document activities were obtained in the storage phase, so the developer can more easily consider if the retrieved use case and BPEL document are worthwhile.

After the developer specifies the use case for the new system, retrieval of a similar use case in the repository is done as follows:

1. For each step in the new use case, assign a key in the same way as keys were assigned in step 5 of the storage phase.
   1.1. Parse each use case step to obtain the values for the following template: "[delay-specification] [condition-statement] [determinant] entity operation-reference".
   1.2. For each use case step, do the following:
      1.2.1. If there is a delay-specification, first consider only that part. Assign the delay-specification term (such as "*AFTER*") as the keyword. For the rest of the use case step, recursively apply step 1.2.
      1.2.2. If there is a condition-statement ("*IF* condition *THEN*"), first consider only that part. Assign "*IF*" as the keyword. For the rest of the use case step, recursively apply step 1.2.
      1.2.3. If the entity is the System under Design (or an attribute of it), assign "*SUD* operation-reference" as the keyword.
      1.2.4. If the entity is the Primary Actor or a Participant, assign "*PA* (or *PART*) operation-reference" as the keyword.
2. Search for a similar use case using the keys generated in the previous step and the keys assigned to the use cases in the repository. Specifically, SUCABS compares the keys for a given pair of use cases, and assigns scores according to how similar two keys are for pairs of use case steps as follows:
   2.1. If the keys completely match, assign 10 points.
   2.2. If the key for the new use case is included within a key for one of the steps in an old use case, assign 10 points. This is to handle cases where the correlation between use case steps and BPEL activity is actually 2 to 1, such as for *invoke* activities.
   2.3. If the first term of the key matches (e.g., the entities are both *SUD*, *PA*, or *PART*), and the verb of the operation-reference is a synonym according to Wordnet [12], assign 7 points.
   2.4. If the first term matches but the verb of the operation-reference neither matches nor is a synonym, assign 2 points.
3. Divide the total points by the number of steps in the new use case, resulting in the point for a particular use case to be between 0 and 10 points.
4. The result is shown to the developer for his/her consideration.

The scores in step 2 were obtained through trial-and-error. More work needs to be done to refine the types of scores possible and their values.

5. Discussion

We first describe a usage scenario and then discuss SUCABS.

5.1. Usage Scenario

Since pairs of use cases and corresponding BPEL documents need to be stored first, we first describe a storage scenario, and then describe a search scenario.

5.1.1. Storage

After the developer finishes developing an SoS, he/she stores use case and BPEL document in the repository. This is done by first starting SUCABS and specifying the files for the use case and BPEL document that is to be saved. After loading the files, the main window for SUCABS will look as shown in Figure 1. The upper left subwindow shows the source for the use case, and the upper right subwindow shows the source for the BPEL document.

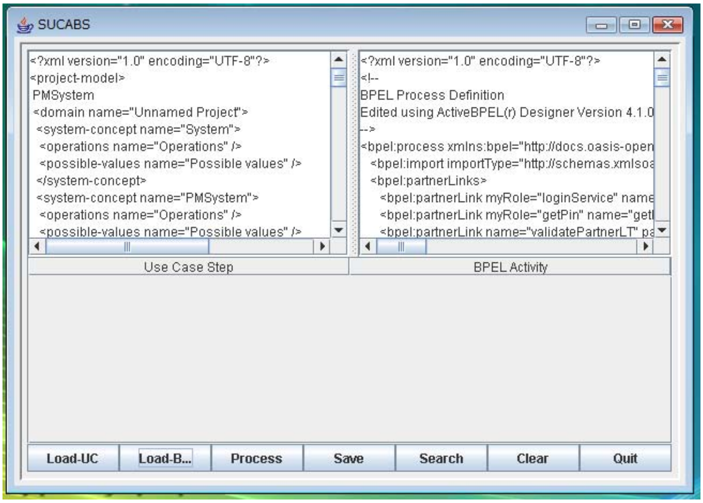


Figure 1. After loading the use case and BPEL files

Then, the user presses the Process button to process the two files and correlate use case steps with BPEL activities. The result is shown in Figure 2. The table shows the correlations. For example, “[User inserts a Card in the card slot]” corresponds to a *receive* activity in the BPEL document, and the two use case steps “[PMSystem asks for PIN, User types her PIN]” corresponds to the *invoke* activity

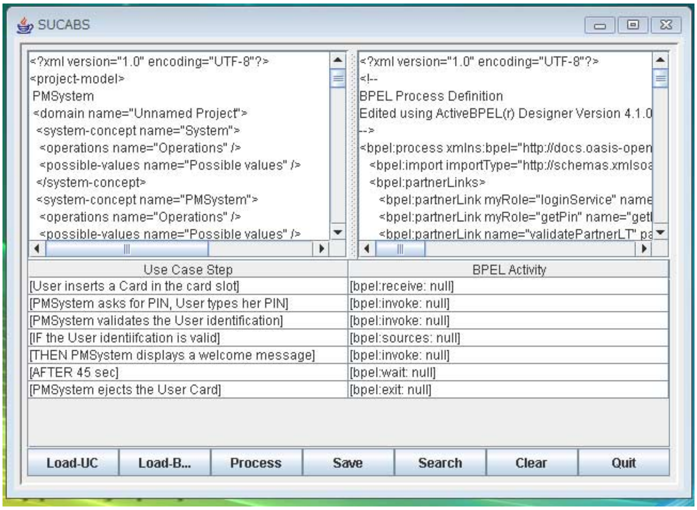


Figure 2. After correlation

At this point, by pressing the Save button, both files, correlation information, and the keys for future searches are saved in the repository.

5.1.2. Search

Suppose that the developer is building an ATM, and wrote the following use case for login with UCEd.

> Title: Identify User
> Description: This use case shows a login procedure to an ATM.
> System Under Design: ATM
> Primary Actor: User
> Steps:
> 1. User inserts a Card in the ATM card reader slot.
> 2. ATM asks the User's Personal Identification Number.
> 3. User writes a PIN.
> 4. ATM checks the User's identification.
> 5. ATM displays an operation menu.

In order to search with SUCABS, the Search button is pressed from the main window. The developer chooses the file that the above use case is saved in. SUCABS then analyzes the chosen file and searches for similar use cases, as was described in section 4.3. This results in a window showing the contents of the above use case (Figure 3; left window) and the main window showing the most similar use case, corresponding BPEL document, and correlation information (Figure 3; right window).

The developer considers the information in Figure 3 to see if the BPEL document can be used in the development of the ATM. If the developer thinks it can be used, then the BPEL document is modified and incorporated as part of the ATM BPEL document.

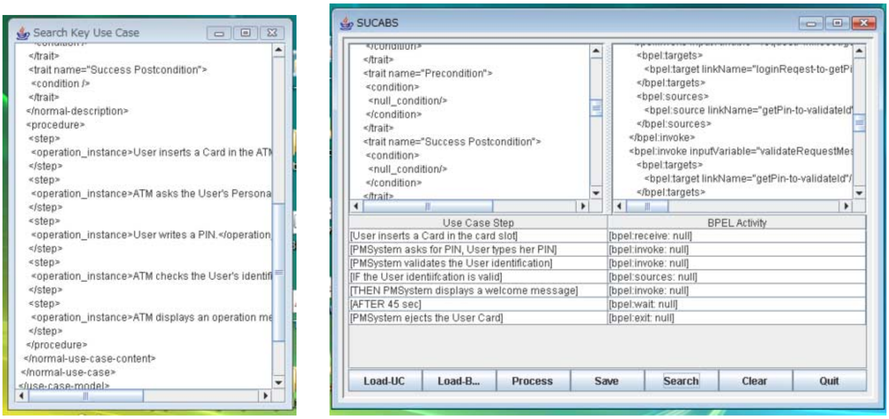


Figure 3. Search result

5.2. Discussion

Our approach correlates use case steps and BPEL activities. One option would have been to stop at correlating a use case with a BPEL document, but we went one step further by correlating use case steps and BPEL activities. We took the extra step so that the developer can actually take into account the actual BPEL activity that is used for a use case step.

Of course, more information can (and should) be provided. For example, the current implementation of SUCABS does not show the Web services that are actually called from a previously developed BPEL document. This is an important piece of information that should be shown in future versions of SUCABS, but we should note that recent trend for the actual determination of the Web services to be invoked is done dynamically at runtime, and not statically at design time.

Our tool assumes that the use case document is made with UCEd. This means that in order to use our approach, the developer may need to change his/her development process to incorporate building use cases with UCEd. Of course, SUCABS can be extended if a different format is preferred. The key point is that whatever format is used, it will need to have steps that describe how a use case is to be carried out.

Contrary to the use case format, BPEL is a standard and thus SUCABS should be able to handle BPEL documents handled by any BPEL design tool. Furthermore, BPEL is widely enough used so that many developers will already be using it for developing SoS. Unfortunately, the reality is that most BPEL tool extends the BPEL standard in some way. SUCABS was tested using BPEL documents generated by ActiveBPEL Community Edition Designer [13]. Thus, there is a possibility that it may not handle BPEL documents from other BPEL design tools.

Other issues include interface. Currently the source files are shown *as is* in XML format. This may not be easily understandable, and thus an interface which shows the BPEL document in a graph style, and also a different more understandable form (such as ordinary text) for the use case document would seem to be beneficial.

6. Conclusion

In this paper, we proposed a tool called SUCABS to support the development of service-oriented systems by suggesting possible BPEL documents that can be reused based on use case information. The main function of SUCABS is to correlate use case documents and BPEL documents, and to search for similar use cases (and their corresponding BPEL documents).

The correlation between a use case document and BPEL document is based on checking each use case step and BPEL activity. The search process is based on checking the similarity of each step in the use case. Wordnet is used to allow for usage of different words but with the same meaning.

Although there have been work on reusing use cases in the development of systems [14] [15] [16], to the best of our knowledge, there have been no work done in reusing BPEL documents through use cases to develop a service-oriented system.

Much work is still necessary. Although we have described some future work in section 5, other future work includes integration of SUCABS with a use case editor and BPEL editor. Currently, SUCABS, UCEd and ActiveBPEL are all separate entities, and the developer needs to transfer files between them. We also need to do some more research on calculating similar use cases. Currently, the scores for similar use case steps were derived from a simple experiment. A more comprehensive experiment needs to be done for validating and refining the scores themselves.

Acknowledgement

This work was supported by JSPS Grant-in-Aid for Scientific Research (C) 18500026.

References

[1] OASIS: "Universal Description, Discovery ,and Integration Version 3.0.2", http://uddi.org/pubs/uddi_v3.htm (2004).
[2] Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara: "Semantic Matching of Web Services Capabilities", Proc. of First International Semantic Web Conference, pp.333-347 (2002).
[3] J. Pathak, N. Koul, D. Caragea, V. Honavar: "A Framework for Semantic Web Services Discovery", Proc. of the 7th ACM International Workshop on Web Information and Data Management (WIDM 2005), pp.45-50 (2005)
[4] Liangzhao Zeng , Boualem Benatallah , Marlon Dumas , Jayant Kalagnanam , Quan Z. Sheng: "Quality driven web services composition", Proc. of the 12th International Conference on World Wide Web, pp.411-421 (2003).
[5] Kuniharu Yamada, Shingo Takada, Norihisa Doi: "Towards Use Case based Software Reuse", Proc. of 8th IASTED International Conference on Software Engineering and Applications (SEA 2004), pp.725-730 (2004).
[6] Diane Jordan, John Evdemon, Alexandre Alves, Assaf Arkin, Sid Askary, et al.: "Web Services Business Process Execution Language Version 2.0"", Public Review Draft, OASIS WSBPEL Technical Committee, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.pdf (2006).
[7] OMG: "OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2", http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF (2007).
[8] Alistair Cockburn: "Writing Effective Use Cases", Addison-Wesley (2001).
[9] Stephane Somé: "Supporting Use Case based Requirements Engineering", Information and Software Technology, Volume 48, Issue 1, pp.43-58 (2006).

[10] Sang Duck Lee, Young Jong Yang, Eun Sook Cho, Soo Dong Kim, Sung Yul Rhew: “COMO: A UML-Based Component Development Methodology,” Proc. of 6th Asia-Pacific Software Engineering Conference (APSEC'99), pp.54-61 (1999).
[11] Remco Dijkman, Stef Joosten: “Deriving Use Case Diagrams from Business Process Model”, CTIT Technical Report, University of Twente (2007).
[12] Wordnet: http://wordnet.princeton.edu/
[13] ActiveBPEL: http://www.activevos.com/community-open-source.php
[14] R. Biddle, J. Noble, E. Tempero: “Supporting Reusable Use Cases”, Proc. of the 7th International Conference on Software Reuse, pp.210-226 (2002).
[15] J. Cybulski, K. Reed: “Requirements Classification and Reuse: Crossing Domain Boundaries”, Proc. of the 6th International Conference on Software Reuse, pp.190-210 (2000).
[16] M. Mannion, H. Kaindl, J. Wheadon, B. Keepence: “Reusing Single System Requirements from Application Family Requirements”, Proc. of the 21st International Conference on Software Engineering, pp.453-462 (1999).

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-453

Towards a knowledge-sharing approach for Information Security Risk Management

Katerina PAPADAKI[a] and Nineta POLEMI[b]
[a]*National Technical University of Athens & Bank of Greece*
[b]*University of Pireaus*

Abstract. Information security risk management (ISRM) is a major concern of organizations worldwide. Although the number of existing ISRM methodologies is enormous, in practice a lot of resources are invested by organizations in creating new ISRM methodologies in order to capture more accurately the risks of their complex information systems. This is a crucial knowledge-intensive process for organizations, but in most cases it is addressed in an ad hoc manner. The existence of a systematic approach for the development of new or improved ISRM methodologies would enhance the effectiveness of the process. In this paper we propose a systematic meta-process for developing new, or improved ISRM methods. We also present the specifications for a collaboration and knowledge-sharing platform supporting a virtual intra-organizational cross-disciplinary team, which aims at improving its ISRM methodologies by adopting the proposed meta-process.

Keywords. Information security, risk management, knowledge-sharing

Introduction

As the world grows more dependent on IT systems and processes, management of information technology (IT) risk becomes a practical necessity [1]. It is the responsibility of each organization to determine the risk management methodology best suited to the specific environment and culture [2]. A methodology created for one organization may be entirely inappropriate for another. Even if they adopt an "of-the self" methodology, organizations need to modify it in order to adapt it in their specific environment. This fact explains the plethora of existing information security risk management methodologies [25].

Furthermore, as conditions change over time in the external and the internal environment of the organization, the risk management methodologies have to evolve accordingly. Experts in risk management recognize the need to monitor the information security risk management process itself and acknowledge that, for companies to rely on their risk management processes, refinement through constant review and updating is critical [3]. Although the development of new or improved information security risk management (ISRM) methodologies is an important organizational knowledge intensive process, the study of existing literature reveals that currently there is not adequate research regarding this issue. In this paper we propose a meta-process, build on the theoretical foundation of Total Systems Intervention [4], for the development of new and improved ISRM methodologies.

Moreover the proposed meta-process is characterized by interdisciplinary knowledge sharing and collaboration, that could be more effectively realized within a Network of Practice [5] for risk management, an intra-organizational cross-functional virtual team of individuals with experiences and knowledge gained from practicing risk management related tasks. We also present our vision for the creation of a collaboration and knowledge-sharing platform that will support the participation in the proposed network.

This paper is structured in the following way: in section 2 we propose a network of practice for ISRM. In section 3 we present the meta-process for ISRM. In section 4 we identify the requirements of a collaborative and knowledge-sharing platform for ISRM, we present the main components of the platform and a possible scenario demonstrating the main functionalities. Finally, we present the main conclusions of the first functionality testing of the platform and also our plans for further improving it in the future.

1. A Network of Practice for ISRM

The process of managing information security consists of all activities, which achieve and maintain appropriate levels of confidentiality, integrity, and availability. Non-repudiation, accountability, authenticity and reliability should be considered appropriately. According to information security standards (ISO/IEC 27001, 2005), a systematic approach to risk management is necessary for the identification of requirements for information security within an organization, as well as its implementation and ongoing administration to create effective information security management system. Risk management analyses in depth what can happen and what damage can occur before defining what should be done (and when) to reduce the damage to an acceptable level. The ISRM process consists of two main elements: risk assessment and risk treatment. The information security risk assessment is composed of risk analysis and risk evaluation. Furthermore, risk analysis is composed of risk identification and estimation (Parker, 2006). These sub-processes are basically the same in all ISRM approaches. Naturally the following question arises: since all approaches consist of more or less the same elements, how are the numerous information security risk management approaches differentiated from each other? Each of these sub-processes can be implemented by using many different techniques and methods. For example there are many risk identification techniques, such as checklists, flow charts, scenario analysis, fault tree analysis etc. Likewise, there is a plethora of qualitative or quantitative approaches for risk estimation (Peltier, 2001). There are various techniques that can be used to determine the depth of risk assessment required in a given instance. Therefore risk assessment begins by identifying the appropriate approach to risk assessment, i.e. identifying methods or techniques of risk assessment that best suit the specific environment and culture of each organization. Following the risk assessment, risk treatment decisions need to be made on whether to avoid, transfer, accept or reduce the risks that have been identified. Risk treatment is clearly a decision making phase, for which there are many techniques and methods, especially from the domain of decision sciences, that can be used. The selection of the appropriate methods for risk treatment is also a very important decision.

The above analysis leads to the conclusion that the selection of the appropriate techniques and methods to be used by an organization in the risk management process

is a very important decision. In addition ISRM is an on-going process during the life cycle of the organization. As the conditions in the internal and external environment of an organization continuously change, traditional risk assessment methods become difficult to apply. For companies to rely on their risk assessment processes, refinement through constant review and updating is considered critical (Parker, 2006). Today, a very important driver behind the need for continuous improvement of the ISRM approaches is the trend towards integrating all the risks of an organization under the same framework. In enterprise wide risk management frameworks like COSO overall business risks must include technology (COSO, 2004). It becomes therefore apparent that areas such as information confidentiality and privacy cannot be assessed in a vacuum (Parker, 2006). The information security risks must be directly linked to the achievement of the business objectives of the organization. Information security researchers emphasize the need for improving existing information security risk management approaches by including methods from the social sciences (Dhillon and Backhouse, 2001; Siponen, 2005; Gerber and von Solms, 2005). The need for continuous improvement of the information security risk management process, by incorporating techniques and methods from the social sciences, requires the active participation of people working in the same organization, but outside the information security function.

Risk management processes are often defined in many functional areas throughout an organization. For example, risk management is a key process in product certification, project management, financial analysis, development of a business strategy, and in information technology and corporate governance. The methods and techniques for risk management in each area are different: balanced scorecards, multiple criteria analysis, simulation, data envelopment analysis, and financial risk measures that help assess risk, thereby enabling a well-informed managerial decision making (Olson and Wu, 2007). People engaged in risk management tasks from different functional areas of an organization should be able to exchange knowledge. This would result in the cross-fertilization, and ultimately the improvement, of risk management methods. Undoubtedly, one of the characteristics of a successful organization is the capability to diffuse knowledge in the right directions among its functional divisions, in a better way than its competitors in the market (Nonaka and Takeuchi, 1995). John Seely Brown and Paul Duguid (2001) introduced the concept of Networks of Practice as a social-practice perspective for addressing various issues of organizational knowledge. The primary reason for the emergence a Network of Practice is the sharing of practice-related knowledge between individuals in order to perform their work. In the case of risk management, people within an organization working in different functional areas but sharing the knowledge of practicing risk management would considerably benefit by participating in a Network of Practice. We envisage a Network of Practice for Risk Management, which cuts horizontally across a vertically integrated organization. By being part of this network people from the information security function can gain access to the knowledge necessary for improving their risk management approach.

The realization of this Network of Practice has to overcome spatial and temporal constraints: we acknowledge that not all participants work in the same place or during the same hours. But since Networks of Practice range from communities of practice to electronic networks of practice (also known as virtual or electronic communities), it is not necessary for the members of the network to actually meet. We have entered the Web 2.0 era and we have in our disposition a variety of services and applications that

enable the creation of virtual communities. Recent research regarding participation in virtual communities reveals significant increase in knowledge sharing and collaboration performance (de Moor and Weigand, 2007; Hsu et al, 2007; Chiu et al, 2006; Yangand and Chen, 2008). A collaboration and knowledge-sharing platform based on these technologies will support the Network of Practice for risk management. More information on this platform is provided in Section 4.

In order to be beneficial for the improvement of information security risk management methodologies, the participation to this network should not have an ad hoc character. It is crucial to identify a meta-process that will ensure the effective sharing of existing explicit and tacit knowledge and also the efficient organization and dissemination of created knowledge.

2. A Meta-Process for ISRM

Although the continuous improvement of information security risk management methodologies is an important issue for all organizations, limited information is available in information security literature regarding the process of developing these methodologies or of improving existing ones [22][23][24]. A reason for the shortage of evidence regarding the process of developing or improving ISRM methods is that organizations prefer to keep this information confidential.

A more pragmatic reason is that the development of these methods within organisations is rather an ad hoc process than a systematic one. The process of developing or improving these methods generates new knowledge about information security risk management, which constitutes valuable organizational intelligence. Therefore, it is very important to have a systematic process, or rather a meta-process, which ensures that the acquired knowledge will be elicited, shared and managed appropriately [8].

In our effort to propose a meta-process for ISRM we used the Total Systems Intervention (TSI) theory [4]. The reason for this selection is best described by Midgley [9]: "…many situations are so complex that a variety of methods are often needed to tackle them adequately. Therefore it is more useful to think in terms of methodology design than simple choice between "of-the-shelf" methodologies – the concept of "creative methodology design" - the creative design of methods."

We draw upon the meta-methodology of TSI in order to distinguish between three modes of the risk management process:

1. **Performing information security risk management mode.** In this mode the actual performance/practice of risk management process, as we know it, takes place. This means that the selected risk management approach, i.e. the techniques considered appropriate by the organization, is applied in real situations.
2. **Criticizing information security risk management mode.** This is a mode that functions as a post-mortem analysis of the "performing" mode. The objective of this mode is to evaluate the experiences from applying the methods. In this mode the participants comment on how well the methods contributed to the overall success of the risk management and identify problems they encountered.
3. **Reviewing information security risk management mode.** Having identified certain problems in the criticizing mode, in the reviewing mode we aim at

finding solutions. In this mode the methods or techniques that have not performed adequately will be improved or replaced by other methods. So the participants identify the requirements of the new method, propose alternative methods that satisfy the requirements and finally they choose a method that is considered more appropriate.

In practice only the first mode (practicing) has been the subject of research. The other two modes (criticizing and reviewing) although they exist they are not explicitly acknowledged or addressed in a systematic way.

Most practitioners of ISRM would probably consider that criticizing and reviewing the risk management process would most realistically performed by researchers, who are more likely to have the time and opportunity to invest in the exploration of a methodology's theory and practice prior to the use of those methodologies in performing the process. Our answer to this view is the creation of the Network of Practice [5], i.e. an intra-organizational cross-functional team of people with practical experience participating in the information security risk management meta-process. Risk management processes are often defined in many functional areas throughout an organization. For example, risk management is a key process in product certification, project management, financial analysis, development of a business strategy, and in information technology and corporate governance. The methods and techniques for risk management in each area are different: balanced scorecards, multiple criteria analysis, simulation, data envelopment analysis, and financial risk measures that help assess risk, thereby enabling a well-informed managerial decision making [32]. People engaged in risk management tasks from different functional areas of an organization should be able to exchange knowledge. This would result in the cross fertilization, and ultimately the improvement, of risk management methods. Unfortunately in large organizations these people do not have the time or the opportunity to cooperate and exchange experiences in order to improve their methods. This causes what organizational and knowledge management scientists call "stickiness of knowledge" [14].

In order to overcome the time and distance constraints and to achieve a wider participation the prerequisite is the existence of a platform that will support the whole effort. We have entered the Web 2.0 era and we have in our disposition a variety of services and applications that enable the creation of virtual communities [30] [31]. A collaboration and knowledge-sharing platform based on these technologies will support the Network of Practice for risk management. Recent research regarding participation in virtual communities reveals significant increase in knowledge sharing and collaboration performance [26][27][28][29].

In order to come up with the requirements for such a platform we need to identify the characteristics of the three modes of ISRM meta-process:

- **Iterative process:** the three modes (practicing, criticizing, and reviewing) are performed in an iterative manner. When practicing risk management internal or external factors create inconsistencies that require adjustments possibly by the introduction of a new method. Moreover it is an on-going process that takes place in parallel with all other activities during the life cycle of the organization.
- **Consensus decision-making/problem-solving process**: the three modes of risk management require the participation of all stakeholders that collaborate in order to make decisions. During practicing of risk management the involvement of all interested parties ensures the accurate assessment of the most important risks and the right decisions on risk treatment. During the

criticizing the participation ensures that the maximum number of issues will be surfaced and decisions will be taken regarding which of these will be addressed in the reviewing mode. Finally during the review mode the participants decide on the method that is more appropriate for the issues identified in the criticizing mode. Consensus is needed in order for the decisions to be accepted by all stakeholders.

- **Interdisciplinary knowledge sharing**: information security risk management is not just a technical issue. On the contrary it is an issue with multiple key dimensions (e.g. business, economic, culture, legal, politics, standards, technology) that need to be taken into consideration. . Clearly the existence of many viewpoints ensures a holistic approach towards information security. The prerequisite for such an approach is to have people with different backgrounds participating in the Network of Practice for ISRM.

3. A Risk Management Knowledge Sharing and Collaboration Platform

The characteristics of risk management meta-process, already described in this paper, show that it is a creative, iterative decision making process realized by the collaboration of experts from different disciplines. Knowledge in this risk management meta-process is diverse and steadily growing. Improved use of this knowledge is the basic motivation for knowledge sharing in risk management. The potential limitations of knowledge sharing notwithstanding [4][15][16], we believe it is crucial to have tools that would assist participants in their knowledge-intensive tasks, by enabling them to discover, share, and manage knowledge. Therefore in order to identify the desired properties of a risk management knowledge sharing and collaboration tool we studied best practices for knowledge sharing from the knowledge management domain [17][18][19].

Based on the characteristics of the proposed ISRM meta-process and best practices from knowledge management literature we conclude that a knowledge sharing and collaboration platform must meet the following requirements:

Support for ISRM knowledge codification. Participants will be able to find relevant risk management knowledge. A codification strategy probably works best for certain types of knowledge that is not expected to change frequently. Participants can then easily retrieve methods and best practices that have proven themselves in the past, and reuse them accordingly.

Support for ISRM personalization. Because risk management meta-process is consensus decision making, knowledge is not always immediately ‘stable’ enough to codify, because until consensus has been reached, decisions could change. For such knowledge, a personalization strategy could prove useful to enable participants to find who knows what. Furthermore personalization techniques are also valuable to support the discussions and negotiations between stakeholders.

Support for collaboration. Because risk management meta-process is consensus decision-making, a knowledge-sharing tool should explicitly support collaboration between different users. This property enables the active involvement of all important stakeholders in the decision making process.

We are currently working on knowledge-sharing and collaboration platform that will support a Network of Practice for information security risk management. In this section, we present the main vision of this platform. The following central features are

incorporated in the platform in order to allow codification and personalization of risk management knowledge, and to enable collaboration between stakeholders:

Forum. A forum is employed to allow participating members to easily communicate and collaborate. As a result, other stakeholders can quickly acquire information about the current status of the discussion. One important motivation for people to participate in forums is the ability to create a community feeling [20]. To foster communication between stakeholders, and to motivate them to share risk management knowledge, such a community feeling is essential.

Text mining. A forum is a typical personalization approach in which a lot of unstructured information is stored. However, potentially relevant risk management knowledge is much more valuable if it is codified as a reusable asset. Text mining techniques, see e.g. [21], are employed to enrich unstructured risk management knowledge present in the platform.

Repositories. To store reusable knowledge assets the platform has three repositories. The results repository contains the results from practicing risk management. The experience repository captures the criticizing part, i.e. the open issues or the problems identified in risk management methods.

E-mails. E-mails are employed to push relevant risk management knowledge to certain stakeholders, or to notify subscribed stakeholders if new knowledge emerges. E-mails are complementary to the more traditional pull mechanism of using the repositories. Users can subscribe to certain topics of interest and get updated without having to search the platform themselves, which is a lightweight approach to share risk management knowledge among relevant stakeholders.

Expert finding. Similar to the 'yellow pages' concept, the expert finding facility allows users to easily find colleagues based on experience, interests or projects on which they work. By connecting people in the risk management meta process, we increase "team building" and "group feeling" within the organization, and foster discussions that can result in higher quality solutions.

In the following scenario, the interplay between the *forum*, *text mining*, *e-mails* and the *repositories* is apparent demonstrating the desired functionality of the platform. Members of the information security function perform a routine risk management for one of the organization's information systems. All the data from this process are stored in the *results repository* of the knowledge-sharing and collaboration platform.

After the completion of this process the team members reflect on the usage of methods in order to identify any problems. One of the team members has an objection to one of the methods used in risk assessment (e.g. the method used for the assessment of the impact of risks) and thinks that an improvement of the method should be considered. The member uses the *text mining* service of the platform in order to see if this issue has been discussed in the Network of Practice *forum* in the past. The member finds out the past discussions and the reasons for selecting the particular method. The member thinks that the method should be improved and writes a proposal including arguments. The team member inserts this proposal to the *experiences repository* of the platform and also notifies the leader of the Network of Practice by e-mail. The *experiences repository* acts like a logbook of all the proposals for improvement of various risk management methods. The leader of the Network of Practice decides the priority by which proposals for improvement are to be discussed.

The leader asks the members of the Network to propose solutions within a specified time period. For each of the proposed solutions a discussion thread is started in the *forum*. The Network leader asks the members to provide their comments on the

proposed solutions specifying a deadline. All members and the leader participating in the discussions are informed by *e-mails* every time something new is added to the discussions. Members provide arguments for or against the solution. After the given deadline has expired the leader reviews the discussions and the Network members decide on the solution that will be selected. The selected method is then inserted in the *best practices repository.*

4. Conclusions and Further Research

So far we have successfully tested the functionality of the platform, within a limited Network of Practice for risk management, by using it in a laboratory setting for improving an ISRM method by introducing fuzzy multiple attribute group decision-making methods [8]. Extensive testing of the platform in real cases is within our future plans. Participants in the aforementioned testing concluded that the current version of the platform conforms to most of desired properties of a risk management collaboration and knowledge-sharing tool. First of all, it supports ISRM knowledge codification (repositories, text mining), and ISRM knowledge personalization (forum and expert finding), making it a hybrid ISRM knowledge-sharing environment.

Ultimately, the platform enhances collaboration between the participants in the Network of Practice. Additionally, if the organization ensures that users perceive a certain degree of freedom in using the platform, the platform is inherently descriptive in nature as well. Furthermore, the platform supports persistent and visible reputation tracking. People participating in the creation of new risk management knowledge, for example by sharing information or by publishing content, are rewarded by gaining a certain reputation.

In the future, we envision features to make the platform more appealing, for example by supporting flexible, personalized and secure access for all stakeholders through a personal start page. Such a start page allows for role-specific content views so that users can indicate which mode, method, or people are interested in.

References

[1] Symantec: IT Risk Management Report 2: Myths and Realities. Available online: http://eval.symantec.com/mktginfo/enterprise/other_resources/bit_risk_management_report_2_01-2008_12818026.en-us.pdf (2008)
[2] ISO/IEC 27005: Information Technology – Security Techniques – Information security risk management. Committee Draft (2004)
[3] Parker, X.L.: Information Technology Audits. CCH, USA (2006)
[4] Flood, R. L., and Jackson, M. C.: Creative Problem Solving: Total Systems Intervention. Wiley, Chichester (1991)
[5] Brown, J. S. & Duguid, P.: Knowledge and organization: A social-practice perspective. Organization Science, Volume 12, Issue 2 (2001) 198-213
[6] ISO/IEC 27001:2005: Information technology – Security techniques – Information security management systems – requirements. (2005)
[7] Peltier, T.R.: Information Security Risk Analysis. Auerbach (2001)
[8] Papadaki, K., Polemi, D.: Towards a systematic approach for improving information security risk management methods. In Proc. 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC) (2007)
[9] Midgley, G.: Developing the Methodology of TSI: From the Oblique Use of Methods to Creative Design. Systems Practice, Volume 10, Issue 3 (1997) 305-319

[10] COSO (Committee of Sponsoring Organizations of the Treadway Commission): Enterprise Risk Management – Integrated Framework (2004)
[11] Dhillon, G. and Backhouse, J.: Current Direction in IS Security Research: Towards Socio-Organizational Perspectives. Information Systems Journal, Volume 11 (2001) 127-153
[12] Siponen, M.: Analysis of modern IS security development approaches: towards the next generation of social and adaptable ISS methods. Information and Organization, Volume 15 (2005) 339–375
[13] Gerber, M. and von Solms, R.: Management of risk in the information age. Computers & Security, Volume 24 (2005) 16-30
[14] Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company. Oxford University Press, Oxford, UK (1995)
[15] Ghosh, T.: Creating Incentives for Knowledge Sharing. Technical report, MIT Open Courseware. Sloan school of management, Cambridge, Massachusetts, USA (2004)
[16] Haldin-Herrgard, T.: Difficulties in Diffusion of Tacit Knowledge in Organizations. Journal of Intellectual Capital, Volume 1, Issue 4 (2000) 357–365
[17] Hansen, M.T., Nohria, N., Tierney, T.: What's Your Strategy for Managing Knowledge? Harvard Business Review, Volume 77, Issue 2 (1999) 106–116
[18] Desouza, K.C.,Awazu, Y., Baloh, P.: Managing Knowledge in Global Software Development Efforts: Issues and Practices. IEEE Software, Volume 23, Issue 5 (2006) 30–37
[19] van den Brink, P.: Social, Organization, and Technological Conditions that Enable Knowledge Sharing. PhD thesis, Technische Universiteit Delft (2003)
[20] Nardi, B.A., Schiano, D.J., Gumbrecht, M., Swartz, L.: Why We Blog. Communications of the ACM, Volume 47, Issue 12 (2004) 41–46
[21] Fan, W., Wallace, L., Rich, S., Zhang, Z.: Tapping the Power of Text Mining. Communications of the ACM Volume 49, Issue 9 (2006) 77–82
[22] Armstrong, H.: Managing information security in healthcare – an action research experience. In Proc. 16th Annual Working Conference on Information Security (2000)
[23] Butler, S. and Fischbeck, P.: Multi-Attribute Risk Assessment. In Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS) (2002)
[24] Stamatiou, Y., Skipenes, E., Henriksen, E., Stathiakis, N.,Sikianakis, A., Charalambous, E.,Antonakis, N., Stølen, K., den Braber, F.,Soldal Lund, M., Papadaki, K., Valvis, G.: The CORAS approach for model-based risk management applied to a telemedicine service. In Proc. Medical Informatics Europe (MIE'2003) IOS Press (2003) 206-211
[25] ENISA (European Network and Information Security Agency): Risk Management: Implementation principles and Inventories for Risk Management/Risk Assessment methods and tools, Available online: http://www.enisa.europa.eu/rmra/files/D1_Inventory_of_Methods_Risk_Management_Final.pdf (2006)
[26] de Moor, A. and Weigand, H.: Formalizing the evolution of virtual communities. Information Systems, Volume 32, Issue 2 (2007) 223-247
[27] Hsu, M., Ju T., Yen, C., Chang, C.: Knowledge sharing behavior in virtual communities: The relationship between trust, self-efficacy, and outcome expectations. International Journal of Human-Computer Studies, Volume 65, Issue 2 (2007) 153-169
[28] Chiu, C., Hsu, M., Wang, E.: Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories. Decision Support Systems, Volume 42, Issue 3 (2006) 1872-1888
[29] Yangand, S., Chen, I.: A social network-based system for supporting interactive collaboration in knowledge sharing over peer-to-peer network. International Journal of Human-Computer Studies, Volume 66, Issue 1 (2008) 36-50
[30] Jablonski, S.: Guide to web application and platform architectures. Berlin Springer (2004)
[31] Alonso, G.: Web services: concepts, architectures and applications. Berlin Springer (2004)
[32] Olson, D., Wu, D.: Enterprise Risk Management. World Scientific Publishing (2007)

This page intentionally left blank

Program Understanding, Programming Knowledge, Modeling Programs and Programmers

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-465

Design Patterns Identification Using Similarity Scoring Algorithm with Weighting Score Extension

Martin CICHÝ[1], Jaroslav JAKUBÍK
Faculty of Informatics and Information Technology,
Slovak University of Technology, Bratislava

Abstract. The identification of used design pattern as a part of the reengineering process provides useful information about the structure of a system. One of the existing pattern detection methodologies is detection based on similarity scoring algorithm proposed by Nikolaos Tsansalis. In this paper we propose a modification of the approach, in which we evaluate each observed part of the design pattern separately in the relation to the essence of the design pattern. This leads us to weighting of the observed structural parts of a pattern, so the more important structural parts have higher contribution to the calculated score.

Keywords. design patterns identification, similarity scoring algorithm, design patterns, software reengineering

1. Introduction

Patterns are currently used in almost all software development phases (for example analysis patterns, architecture patterns, design patterns, test patterns). A pattern describes an idea or the best practice used in many projects. Design patterns provide solutions to recurring design problems. Design patterns as well as any other patterns are traditionally described in pattern catalogues (for example [1, 2, 3]) with semiformal description. This description contains context, problem, solution and results.

In some projects, the usage of design patterns has appropriate documentation, but many projects miss this kind of documentation. During service and support phase in software development process, this kind of documentation is very important for system maintenance and extension. Maintenance of object-oriented system without appropriate documentation is difficult. With appropriate documentation, a service team can extend the existing system quickly, fix new errors and provide good quality service. The problem can occur when the usage of design patterns documentation is missing. Therefore, the major task for the service team is to design recover, build higher level of

[1] Corresponding author: Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia; E-mail: cilaman@gmail.com.

abstraction from source code, understand the design and architecture of a system. Knowledge about applied design patterns leads to better understanding of design problems solved by software designers, architects when designing software architecture. Understanding of the design problems is a necessary step towards informed changes of the system and of its architecture [4].

The documentation of a design can be also created and updated for an existing software system using reengineering process. Software reengineering can be done manually or by using separate software tools and utilities. There are many software tools for creating UML models in reengineering process, commercial or not (for example Rational rose, Enterprise architect, Rational XDE, Together). At the moment, these reengineering tools do not provide information about used design patterns (based on source code). For the extraction of design pattern information specific methods and tools must be used (for example [4, 5, 6]).

Methods for pattern identification use wide range of IT technologies for getting results in specified time and quality. There are methods which use static, dynamic or both kinds of analysis by:

- mathematical methods (for example graphs, matrices)
- mathematical statistics,
- logic or
- neuron nets.

In this paper we propose a method for design patterns identification based on similarity scoring algorithm. We introduce mechanism called feature weighting for a better design pattern identification. We also present a short case study with the comparison of aboriginal and modified method.

Section 2 summarizes similar methods for design patterns identification. We are mainly concerned with the methods using bit – vector algorithm and similarity scoring algorithm. Section 3 includes the description of extended method using similarity scoring algorithm. In section 4 results and comparison of extended and aboriginal method is described. Finally, section 5 concludes and introduces future work.

2. Related Work

In this section we introduce some of approaches in design pattern detection and identification. There are different structural matching techniques (for example [4, 5, 6]). In this section we briefly describe two of them. The first analyzed method is based on bit – vector algorithm and string representation, the second one uses graph comparison and matrix representation.

2.1. Bit – Vector Algorithm

Authors in [6] presented an adaptation of bio-informatics bit - vector algorithms to problem of design pattern identification. The identification of a design pattern consists of parallel traversing of a program and a design pattern. During the traversing of a program, the entities, that match entities in the design pattern in structure and in organisation, are recorded. Design patterns identification is a combinatorial problem, requiring all possible combinations of entities to be compared against a design pattern.

The authors created iterative bit – vector algorithm for identification of exact or approximate coincidence between program and design pattern represented by string. Concrete design pattern/system string representation contains classes (class name) and relations between classes (creation, specialisation, implementation, use, association, aggregation, composition). In general, the length of the string representing a design pattern is short, less than 20 tokens, while the length of a string representing a program might vary, depending on the size of the system to analyse, usually thousands of tokens.

A string is converted to a set of bit vectors (characteristic vector) (for example if class C occurs in string, 1 occurs in bit vector for C, otherwise 0). Characteristic vectors are used for finding the entities playing a role in design pattern. Iterative bit – vector algorithm iteratively reads triplets of tokens (roles) in the design pattern string representation and associates program entities to the roles by resolving a unification-like problem using the characteristic vectors.

The process of design pattern identification described in [6] is composed from following specific tasks:

- software models and design pattern models are converted into digraphs (A model of design pattern or software system is a graph where classes are vertices and relationships between classes are oriented edges. If there exists more than one identical relationship between two same vertices, only one edge in graph is kept.)
- digraphs are converted to Eulerian digraphs (Digraph, in general, is not Eulerian digraph if it does not contain Eulerian circle. Graph conversion consists of adding dummy edges between vertices with unequal in-degree and out-degree.)
- Eulerian digraphs are converted to strings (Eulerial digraph contains an Eulerian circuit, which is a cycle traversing each edge exactly once.)
- modified iterative bit-vector algorithm on the string representations is applied to identify exact and approximate occurrences of the design pattern.

Although the entire process is optimized for quick design pattern identification, the representative strings of large software systems are so long that the identification of design pattern can be time consuming.

2.2. Similarity Scoring Algorithm

The work that we present in this paper is built on the ideas of [5] where the author presents design pattern detection methodology based on similarity scoring algorithm. Bondel in [7] defines an iterative algorithm for calculating the similarity between vertices of two different graphs by similarity matrix.

In the context of design pattern detection, the similarity scoring algorithm is used for calculating similarity score between a concrete design pattern and analyzed system. Let GA(system) and GB(pattern) be two directed graphs with NA and NB vertices. The similarity matrix S is defined as an NB x NA matrix whose entry SIJ expresses how similar vertex J (in GA) is to vertex I (in GB) and is called similarity score between two vertices (I and J).

In [5] authors define a set of matrices for describing specific (pattern and system) features (for example associations, generalizations, abstract classes). For each feature, a concrete matrix is created for pattern and for system, too (for example association matrix, generalization matrix, abstract classes matrix). This process leads to a number

of similarity matrices of size NB x NA (one for each described feature). To obtain overall picture for the similarity between the pattern and the system, similarity information is exploited from all matrices. Different features are equivalent in the process of creating final similarity matrix. To preserve the validity of the results, any similarity score must be bounded within the range <0, 1>. Therefore, individual matrices are initially summed and the resulting matrix is normalized by dividing the elements of column i (corresponding to similarity scores between all system classes and pattern role i) by the number of matrices (ki) in which the given role is involved.

Using this algorithm for whole large system will be time consuming (long time process).

The process of design pattern identification includes:

- creating one matrix NxN for one feature, where N is number of classes in system, in pattern,
- comparing matrixes for each feature between pattern and system,
- computing final similarity matrix.

Methodology defined in [5] defines whole procedure for design pattern detection and identification including search space constrains. Defined methodology contains set of predefined assumptions and specific steps for constraining design pattern search space by subsystem separation. In general, each subsystem contains only one pattern.

- reverse engineering for system under study (Each systems characteristic is represented as matrix n x n, where n is number of classes.)
- detection of inheritance hierarchies (The creation of inheritance trees is based on all kinds of generalization relationships. Multiple inheritance cannot be modelled with simple tree, so if class C has multiple parents A, B, it occurs in multiple inheritance trees with A and with B.)
- construction of subsystem matrices (The creation of subsystems is based on inheritance trees and design pattern characteristics. If a design pattern contains one inheritance hierarchy (for example Composite, Decorator), one inheritance tree equals to subsystem. If a design pattern contains more inheritance hierarchies, the subsystems are formed by combining all system hierarchies.)
- application of similarity algorithm between subsystem matrices and the pattern matrices (Normalized similarity scores between each pattern role and each subsystem class are calculated.)
- extraction of patterns in each subsystem (Usually one instance of design patterns is present in each subsystem.

3. Designed Changes and Approach Extensions

Main drawback of the methodology proposed by Tsansalis in [5] was a fact, that all of the structural features like presence of inheritance, presence of an abstract class or interface and others, were treated equally. This resulted in two facts:

1. In case, that the instance of the design pattern has some modifications, some of the structural features can not be identified. This leads to the lower similarity score and as a result the instance could be overlooked.
2. Lowering the threshold score, which represent the decision point between identifying a design pattern and denying of a structure which is not a design

pattern, could lead us to incorrect identification of structures, that are similar to the structure of a design pattern but are not design pattern instances.

The solution to these two problems is the application of weighting of the structural parts of the pattern, so the more important parts, in relation to the essence of the design pattern, play more important role in the process of calculating of the final score. The idea of the weighting is based on a consideration, that every design pattern has some structural features that are essential for the pattern and some other structural features, which are in most cases domain specific. [8] These other structural features are also a part of the pattern structure, but their modification or absence has lower impact on the essence of the pattern.

For example, when we take the Composite pattern as described in [1], we can observe structural features such as the presence of an abstract class (or an interface), the presence of inheritance between Component class and Composite class and the presence of collection method invocations in the Composite class, which represents the presence of aggregation between Composite and Component classes. Because we look at the Composite pattern as at a structure, that should provide a way of designing composite, tree-like structure, we can assume that the presence of collection methods invocation in the Composite class is important to the essence of the pattern, because this feature forms the composite structure.

Based on these considerations we can divide the structural features of design patterns into two major groups:

1. Features, which form the essence of the design pattern. These features are basic for the design pattern and therefore they should be domain invariable, so they should appear unmodified in every instance of the pattern. In the process of similarity score calculation features from this group should be counted with higher weight.
2. Features, which help top up the structure of the design pattern, as it is described by the catalogue (see [1]). Into this group we can include structural features with a higher rate of domain specific modifications and therefore these features can absent in the pattern structure or can be modified in some way. As absence or modification of these features does not affect the essence of the pattern, we should count them with a lower weight, lowering their importance for the final similarity score.

The application of the weighting based on the separation of the features into two groups of importance has two expected results:

- Lower rate of mistaken identifications of parts of the analyzed system, which look like a design pattern instances, but in fact are not real instances.
- At a higher threshold score we should be able correctly identify more design pattern instances, because the methodology proposed by this paper should be less sensitive to the modification of parts of the design pattern, by which we expect a higher rate of domain specific modifications, or even their absence.

3.1. Implementation

Based on the consideration about weighting of the features, we have modified the existing tool, created by Nikolaos Tsansalis as an implementation of [5], that uses similarity scoring algorithm (Figure 1) by adding PatternFeatureResolver.

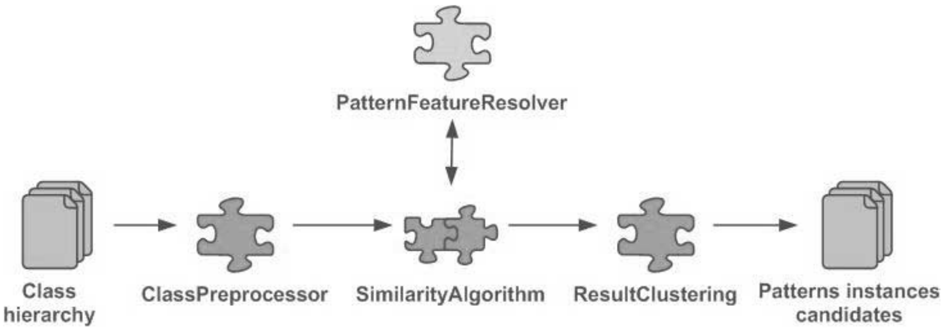


Figure 1: Conceptual tool overview with highlighted modified parts.

The implemented modification consists of three main parts:

- Attributes for storing of the weight. For each observed structural feature of a pattern we have added an attribute used for storing of the weight of the structural part.
- Modification of the class used for construction of the matrices representing the pattern and used in the process of similarity score computation. As the creation of the matrices in the original implementation is hard-coded, we had to add also the initial values of the weights for each observed part of each pattern.
- Modification of the similarity scoring algorithm. We have added a multiplication, which is applied on each partial similarity score of each structural part. As we multiply the partial score with the weight of the part, we can increase or decrease the impact of the partial score on the final similarity score.

4. Case Study

If we want to set the proper weight for each feature of each pattern we have to analyze these patterns and look for their essence. As a case study we have chosen two similar patterns – Composite and Decorator (see [1]).

4.1. Composite Pattern

The composite pattern could be described by the following structural features:

- Presence of an abstract class.

- Invocation of abstract methods inherited from the abstract class.
- Collection methods invocation.
- Presence of generalization.

As mentioned in section 3, the presence of collection methods invocation is the most important structural feature. Absence of this feature could seriously affect the aim of the pattern, which is the creation of composite structures. Based on this consideration, we can assign weights, as is depicted in table Table 1.

Table 1: Weighted features for the composite pattern

Structural feature	**Weight**
Collection methods invocation	Higher
Invocation of abstract methods inherited from the abstract class	Lower
Presence of an abstract class	Lower
Presence of generalization	Lower

4.2. Decorator Pattern

The decorator pattern could by described by the following structural features:

- Presence of an abstract class.
- Invocation of abstract methods inherited from the abstract class.
- Presence of association.
- Presence of generalization.

As the essence of this pattern is decorating instances of a component by adding of additional functionalities to them, we can assume that the presence of association and also invocation of inherited methods are features, which are important for the essence of the pattern.

Presence of association between the Component and the Decorator classes tells us, that the Decorator class contains only one instance of the Component class and that's the instance which should be decorated. Invocation of methods inherited from the Component class indicates, that also the original functionality defined for the decorated instance is executed. This leads us to adjustment of weights as is depicted in table Table 2.

Table 2: Weighted features for the composite pattern

Structural feature	**Weight**
Presence of association	Higher
Invocation of abstract methods inherited from the abstract class	Higher
Presence of an abstract class	Lower
Presence of generalization	Lower

4.3. Application

We have tested the proposed methodology on JHotDraw 6.0 beta library, which represents a large scale system, using weights as suggested in sections 4.1 and 4.2. Features of higher importance were assigned the weighting coefficient 2.0 and the features of lower importance were assigned the weighting coefficient 1.0.

Table 3 contains results of the original methodology, proposed by Tsansalis in [5]. As we can see, at threshold score of 0,75 the methodology was not able to find any instances of composite pattern and only one instance of decorator pattern. At lower threshold score the methodology has incorrectly identified 5 structures similar to the pattern - false negatives and only one real instance - true positive, which lead to very low recall ratio.

Table 3: Results for JHotDraw library without weighting

	Score = 1,0			**Score >= 0,75**			**Score >= 0,5**		
Design pattern	TP	FN	Recall	TP	FN	Recall	TP	FN	Recall
Composite	0	0	100%	0	0	100%	1	5	17%
Decorator	1	0	100%	1	0	100%	11	0	100%

After application of weighting was the methodology able to identify more instances with less false negative occurrences. At the threshold score of 0,75 was the proposed methodology able to correctly identify single instance of composite pattern and 11 instances of decorator pattern, which is 10 instances more than the original methodology. Also at a lower threshold score of 0,5 was eliminated the incorrect identification and the methodology had identified only real pattern instances (Table 4).

Table 4: Results for JHotDraw library with weighting

	Score = 1,0			**Score >= 0,75**			**Score >= 0,5**		
Design pattern	TP	FN	Recall	TP	FN	Recall	TP	FN	Recall
Composite	0	0	100%	1	0	100%	1	0	100%
Decorator	1	0	100%	11	0	100%	11	0	100%

5. Conclusions

The proposed methodology is a solution to the problem of design pattern identification in a fully automated way, based on the methodology proposed in [5]. We have applied the weighting into the process of similarity score computation, which gave us a possibility to consider the partial results in relation with the essence of the identified pattern and later to affect the impact of the results on the final result of the identification. The modifications, we have made, have brought improvement in two aspects:

- The rate mistaken identifications of structures that look like design pattern is lower than the rate of the original similarity scoring methodology.
- At a higher threshold score we were able to identify more instances of design patterns, which means that the proposed methodology is less sensitive to modifications of design patterns and thus is better at identifying of modified design patterns.

5.1. Future Work

Acquired results suggest improvement due the application of weighting on the original methodology proposed by Tsansalis in [5]. This gives us the opportunity to evolve the

methodology proposed in this paper. There are certain areas of interest, which give us a lot of place for future evolution:

- Analysis of the rest of the design patterns [1]. This would result into creation of a design pattern catalogue which would describe domain independent and domain specific parts of the described patterns.
- Extension of the weighting by adding conditions on structural parts which must be included in the instance of the pattern and on the other side, which can not be included in the instance. Afterwards we could be able to filter the pattern candidates based on the presence of these structural parts.
- Extension of the weighting by creating more than two categories of structural parts (as proposed in this paper). Small grained categorization could bring interesting results by giving us more control of the impact of the features on the final similarity score.
- Extension of the base of the structural features which can the methodology observe. This could result in a better description of the identified patterns.

References

[1] E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, 1st edition, 1994, ISBN 0-201-63361-2.

[2] J. Cooper: Java Design Patterns: A Tutorial. Addison-Wesley Personal Education, Massachusetts, 2000, ISBN 0-201-485394.

[3] D. Alur, J. Crupi, D. Malks: Core J2EE Patterns: Best Practices and Design Strategies. Prentice Hall / Sun Microsystems Press, 1st edition, 2001, ISBN 0-130-64884-1.

[4] Y. - G. Gueheneuc, H. Sahraoui, F. Zaidi: Fingerprinting Design Patterns. In: Proceedings of the 11th Working Conference on Reverse Engineering, WCRE'04, 2004, pp. 172 – 184, ISBN 0-7695-2243-2.

[5] N. Tsantsalis, A. Chatzigeorgiou, G. Stephanides, S. T. Halkidis: Design Pattern Detection Using Similarity Scoring. In: IEEE Transactions on Software Engineering. Vol. 32, No. 11, pp. 896 – 909, 2006.

[6] O. Kaczor, Y. – G. Gueheneuc, S. Hamel: Efficient Identification of Design Patterns with Bit-vector Algorithm. In: Proceedings of the Conference on Software Maintenance and Reengineering, CSMR, 2006, pp. 175 – 184, ISBN 0-7695-2536-9.

[7] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, P. V. Dooren: A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. In: SIAM Review, Vol. 46, Issue 4, pp. 647 – 666, 2004, ISSN 0036-1445.

[8] J. Jakubík: Izolácia všeobecných častí vzoru Composite. In: Objekty 2005, Proceedings of the Conference Objekty 2005. Faculty of electrical engineering and computer science, Techncal University of Ostrava, 2005, pp. 74 – 84. ISBN 80-248-0595-2.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-474

Formulation of Clamshell Diagram and Its Application to Source Code Reading

Takehiko MURAKAWA [1], Takayoshi KAWASAKI, Hirofumi MIZUOCHI and Masaru NAKAGAWA
Wakayama University, Japan

Abstract. When describing an idea using letters and diagrams, the concept becomes clear and it is easier to see what to do to achieve a goal. Among the design creation support tools, we focus attention on the Thinking Process Development Diagram of 6 layers. We refer to it as a Clamshell Diagram, and present 16 principles to constitute the diagram. Then we report the support system for drawing the diagrams using a Web browser. Design creation support tools are also useful for analyzing the existing good or bad instances. We therefore attempt to investigate how to represent a source code by means of Clamshell Diagram and apply to GNU Hello to draw a diagram. Using the diagram representation of source code, we can see the whole-part relationship and the problem-solution relationship at a time.

Keywords. design creation support, schematic diagram, source code reading

1. Introduction

There are lots of pieces of software developed and released for us to enjoy the benefit. In making such software, at present, human resources are unavoidable; although so many methodologies and tools such as Model-Driven Architecture[1] are proposed and applied for supporting the software development, automatic generation of source codes does not seem to be established. When we shift our focus to the upper process of software development or idea generation which does not necessarily relate directly to computer issues, there are quite a number of approaches but we cannot find the "Sunday punch" around them.

The authors worked up interest in the Thinking Process Development Diagram (TPDD)[2]. The inventors first have built up a design support tool for mechanical design, and now they are expanding the scope of application. In this paper, we concentrate on TPDDs made up of 6 layers, refer to them as Clamshell Diagrams (CDs), and present the framework of formulation by introducing the principles. Furthermore, the support system for drawing CDs is informed while the investigation of applying the diagrams to source code reading is reported with a concrete example. The goal of our research is to demonstrate the utility of the diagram for both upper and lower processes.

The outline of this paper is shown in Fig. 1, by means of Clamshell Diagram.

[1]Faculty of Systems Engineering, Wakayama University, 930 Sakaedani Wakayama Japan, E-mail: takehiko@sys.wakayama-u.ac.jp

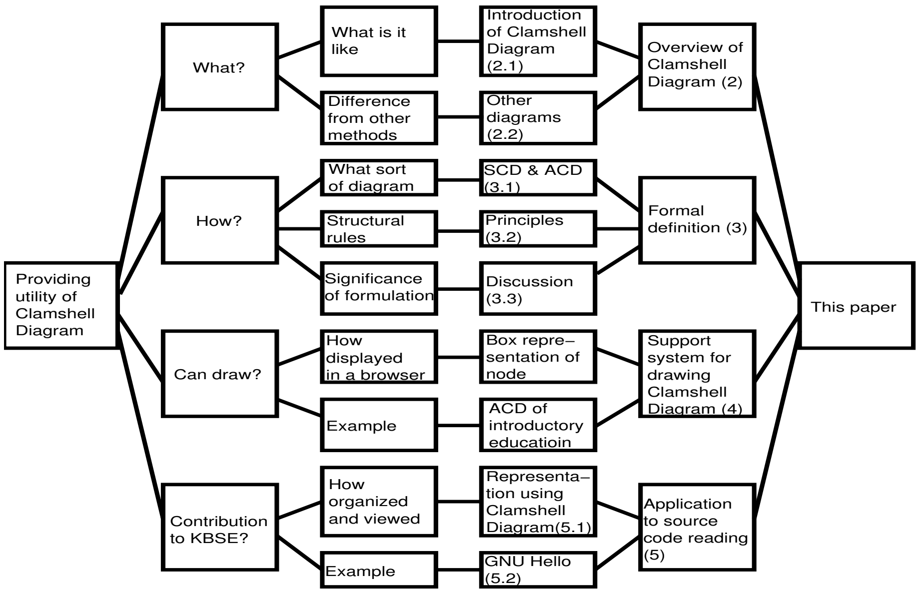


Figure 1. Outline of this paper using Clamshell Diagram

2. Clamshell Diagram and Related Diagrams

2.1. Introduction of Clamshell Diagram

Clamshell Diagram is a diagram of the designer's thought in a certain style. The completed diagram will be represented including two hierarchical relationships and the relationship of issue and resolution. Although the diagrams have been developed in mechanical design, the originators said that they would be available to broader area of design processes, with which we agree.

A standard Clamshell Diagram has two tree structures whose roots are the both edge of the diagram, and each leaf of a tree connects to a leaf of the other tree. The left half of the diagram typically shows the problem to be solved, while the right half expresses the solution. Ideally the diagram is bilaterally symmetric, where the corresponding nodes have the relationship of problem-solution. In other words, an incomplete diagram helps us to find the lack of assertion; if a node appears in the left half but the corresponding one is not observable in the right half, then you will imagine that the means for the specified (sub)goal is unclear; in reverse case, it indicates that the express means does not directed to the end. Such missing nodes are expected to complement after a drawer has a careful look at neighboring nodes and thinks of the hierarchical relationship or the homogeneous elements. The spatial thinking support like this is a remarkable feature of this diagram.

The nodes are labeled and put in position from left to right by ordinary. When drawing a diagram, one begins with the deployment of the left root, and then expands the notions or sentences to the right. In the right half, he or she has to converge the thought, and the right edge is single information, namely the conclusion. Thanks to these structural constraints, we can think of the issue and resolution at a time, and produce the design creatively with a moderate strain.

The authors refer to the diagrams as “Clamshell Diagrams” in this paper, while Mase et al.[2] reported the ancestor diagrams which are called “Thinking Process Development Diagrams.” We rename the diagrams since the word clamshell is the best expressive of the structure; a typical diagram takes the shape of a clamshell where two shells are opened with the joint at both ends. The clamshell is the symbol of unity of two parts in Japan and often mentioned with respect to husband and wife.

2.2. Other Diagrams

KJ Method (Jiro Kawakita’s Method)[3] supplies the way of arranging miscellaneous pieces of information for a subject. Yagishita et al.[4] attempted a quantitative evaluation of the resulting contents. KJ method roughly consists of two steps; noting down each idea on a card, and consolidating the cards. The former step corresponds to the act of creating a node in our supporting system reported in Sect. 4. However, the latter step seems to lack a constructive rule of organizing the ideas. Clamshell diagrams present the clear constructions of nodes as well as the whole-part relationship.

A Mind Map is widely used for expanding a bit of idea[5]. With a single keyword centered, one draws curves in every direction to attach relevant notions. From a view of construction, a diagram is regarded as a directed tree where the centered keyword is the root. The trouble is that, consequently, if a phrase occurs twice or more on the diagram, then they should be separated. Another weak point of Mind Maps is that how widely and deeply the ideas should be expanded lies in the hand of the drawer. Someone might complete a diagram where one direction is further expanded while the others are poor. That is because of wrong caliber of the drawer or because of wrong establishment of the centered subject. The Clamshell Diagram is so clear since the goal should be a condensed single notion. In addition, if the initial subject get to be inadequate in proportion as the nodes, then one can change the centered topic.

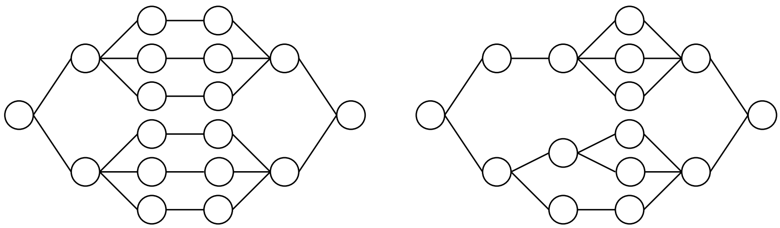
A Cause Effect Diagram has a backbone and small bones which directly or indirectly connect to the backbone. The diagram is also known as “Fishbone Diagram” by its shape, or as “Ishikawa Diagram” after the inventor. While used in the Japanese business community, the diagram is now so popular that an example is found in [6]. That is used for presenting all the factors for a given property or result. The Cause Effect Diagram reads the backward reasoning, which distinguishes it from the Mind Map.

TRIZ is a methodology for creation support based on the patterns derived from enormous number of patents[7]. That has a matrix consisting of some dozens of parameters to clarify the technical problems for the combinations. Schlueter[8] used it to improve a GUI application written in Perl/Tk.

3. Formal Definition of Clamshell Diagram

3.1. Symmetric and Asymmetric Clamshell Diagrams

Among several sorts of Thinking Process Development Diagrams, we are focusing on the diagrams which has exactly six layers of nodes to name them Clamshell Diagrams. If the left and right halves of the diagram form tree structures where the leftmost node and the rightmost node are the roots respectively, and the leaves of the left tree have one-



(a) Symmetric Clamshell Diagram (b) Asymmetric Clamshell Diagram

Figure 2. Two sorts of Clamshell Diagram

to-one relationship with those of the right tree, then the diagram is called a Symmetric Clamshell Diagram (abbreviated as SCD).

SCDs are tractable by using computers and databases, while they are very restricting for us to draw pragmatic diagrams. Such an incommodity urges the authors to think of a moderate diagrams which keep the practical utility; on a new sort of diagram, the left four layers form a tree. Just same as SCD, the right three columns constitutes the tree structure. We call the new sort of diagram Asymmetric Clamshell Diagram, ACD for short. Figure 2 shows the constructional difference between SCD and ACD.

Here we are presenting the principles for defining SCDs and ACDs. Since ACDs comprise SCDs by definition, the principles for ACDs should be properly smaller than those of SCDs. We actually enumerate the principles satisfying the requirement.

3.2. Principles

A Clamshell Diagram is formally an undirected graph $G = (V, E)$, where the following principles hold.

Principle 1: V is a set of nodes and partitioned into six subsets denoted by V_1, V_2, ..., V_6, such that $V = V_1 \cup V_2 \cup V_3 \cup V_4 \cup V_5 \cup V_6$ and $V_i \cap V_j = \emptyset$ if $i \neq j$.

Principle 2: E is a set of undirected links and partitioned into five subsets E_1, E_2, ..., E_5, such that $E = E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5$ and $E_i \cap E_j = \emptyset$ if $i \neq j$.

Principle 3: If $(u, v) \in E_i$ then exactly one of u and v belongs to V_i, while the other is an element of V_{i+1}.

Principle 4: $\#V_1 = \#V_6 = 1$, where $\#A$ denotes the cardinality of the set A.

Principles 1–3 form the fundamental structure of Clamshell Diagrams. When drawing the diagram actually, one would make a list of labels, and then align them (The set V_i corresponds to the nodes of the i-th column from the left in Fig. 2(a).) to fix the connections of the labels. Principle 4 requires exactly one goal of the design target and exactly one piece of information for the solution of realizing the goal on the diagram.

Principle 5: Let i be one of $1, 2, 3$, then

- **(i)** For any $v \in V_{i+1}$, there exists u such that $(u, v) \in E_i$.
- **(ii)** For any $u \in V_i$, there exists v such that $(u, v) \in E_i$.

Principle 6: Let i be either one of $4, 5$, then

- **(i)** For any $u \in V_i$, there exists v such that $(u, v) \in E_i$.

(ii) For any $v \in V_{i+1}$, there exists u such that $(u, v) \in E_i$.

If there exists a node v violating Principle 5(i) or a node u violating Principle 6(ii), then it means that the dominant conception for the node is unclear. If there exists a node u violating Principle 5(ii) or a node v violating Principle 6(i), then it indicates that the subordinate conceptions are lacking.

For any Clamshell Diagrams satisfying Principles 1–6, the sets V_1 and V_2 are a one-to-many relationship. So are the sets V_6 and V_5. However, the consecutive two of the sets V_2, V_3, V_4 and V_5 are many-to-many in general. The following Principles 7–8 will make the relationships one-to-many.

Principle 7: For any $v \in V_{i+1}$ where $i \in \{1, 2, 3\}$, there exists u uniquely such that $(u, v) \in E_i$.

Principle 8: For any $u \in V_i$ where $i \in \{4, 5\}$, there exists v uniquely such that $(u, v) \in E_i$.

Principles 1–8 require the two subgraphs of G, namely $G_L = (V_1 \cup V_2 \cup V_3 \cup V_4, E_1 \cup E_2 \cup E_3)$ and $G_R = (V_4 \cup V_5 \cup V_6, E_4 \cup E_5)$, to be tree-structured. With respect to the number of elements, $1 = \#V_1 \leq \#E_1 = \#V_2 \leq \#E_2 = \#V_3 \leq \#E_3 = \#V_4$ and $\#V_4 = \#E_4 \geq \#V_5 = \#E_5 \geq \#V_6 = 1$ hold.

The rest of this section assumes that the intended Clamshell Diagrams satisfy Principles 1–8. Although it seems to be interesting to investigate the conditions of no intersection without assuming Principles 7–8, that leaves a future work. In addition, consider A is a list (an ordered set) instead of a mere set and $A[p]$ denotes the p-th position of the list where $A[1]$ means the beginning.

Principle 9: Let $i \in \{1, 2, 3\}$, and p, q, x, y be integers such that $(V_i[p], V_{i+1}[x]) \in E_i$ and $(V_i[q], V_{i+1}[y]) \in E_i$. If $p < q$ then $x < y$.

Principle 10: Let $i \in \{4, 5\}$, and p, q, x, y be integers such that $(V_i[x], V_{i+1}[p]) \in E_i$ and $(V_i[y], V_{i+1}[q]) \in E_i$. If $p < q$ then $x < y$.

If a given diagram satisfies Principles 1–10, we are sure to construct the diagram in a natural way, without crossing the links. For the links violating Principles 9–10, you might remove the crossing by reordering the nodes to find another crossing. Such a diagram involves a complicate relationship in it, or inconsistency.

The following three principles define the symmetrical property of the diagram.

Principle 11: For any $u \in V_3$, there exists $v \in V_4$ uniquely such that $(u, v) \in E_3$.

Principle 12: The equation $\#V_2 = \#V_5$ holds.

Principle 13: For any $u_1, u_2 \in V_4$ where $u_1 \neq u_2$, and $v \in V_5$ such that $(u_1, v), (u_2, v) \in E_4$, there exist $w_1, w_2 \in V_3$ and $x \in V_2$ such that $(w_1, u_1), (w_2, u_2) \in E_3$ and $(x, w_1), (x, w_2) \in E_2$ holds.

Principles 7 ($i = 3$) and 11 are the evidence of the one-to-one relationship between V_3 and V_4. Therefore $\#V_3 = \#V_4$. When the diagram satisfies Principles 1–12, $\#E_2 = \#V_3 = \#E_3 = \#V_4 = \#E_4$ holds, in addition to the magnitude relations described in the principles for tree structures.

Principle 13 means that if some nodes in V_4 have the common rightward parent in V_5, then they have the same leftward parent's parent in V_2, which suggests the symmetric property between V_2 and V_5.

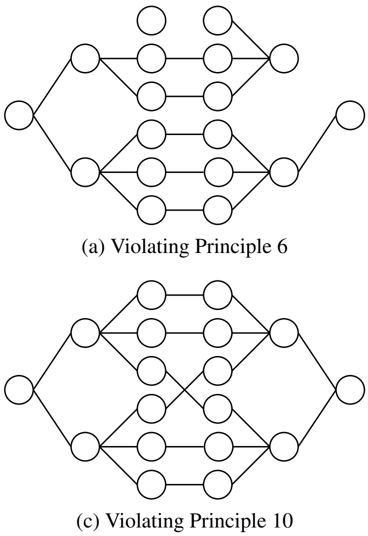


Figure 3. Inadequate diagrams

Principle 14: Let i be one of $1, 2, 3$. For any $u \in V_i$ and $v \in V_{i+1}$ such that $(u, v) \in E_i$, u is a dominant conception of v.

Principle 15: Let i be either one of $4, 5$. For any $u \in V_i$ and $v \in V_{i+1}$ such that $(u, v) \in E_i$, v is a dominant conception of u.

Principle 16: Let i be one of $1, 2, 3$. For any $1 \leq p \leq \#V_i$, $V_i[p]$ and $V_{7-i}[p]$ forms a correspondence relationship of ends and means.

Unlike all the previous principles, it might be impossible to automatically decide whether a given diagram satisfies Principles 14–16, since the semantic analysis for the labels on the nodes is indispensable. Nevertheless, by establishing a database of drawn Clamshell Diagrams to calculate some statistical data on co-occurrence relation among the keywords, we could judge the validity of the connected nodes.

3.3. Discussion

We have axiomatized the well-formed Clamshell Diagrams using principles. They are useful for support system of drawing diagrams. While the diagrams in process of creation are lacking in some nodes or connections, the system must afford to handle such uncompleted diagrams. In addition, the system should give a suggestion to the users on the constitution. Though our support system described in next section does not do so, the principles would be helpful to devise the feature of advice.

Figure3 shows four examples of incomplete or inadequate Clamshell Diagrams.

We should confirm the importance of Principle 13. Any diagram which does not satisfy Principle 13 is ill-formed in both structural and semantic sense, since it indicates the mismatch between the feature and mechanism (or between ends and means). Therefore we conclude that the requirement of SCD is to satisfy Principles 1-13 while that of ACD consists of Principle 13 as well as Principles 1-10.

We describe the rules so that one can judge whether a given diagram compatible with each principles separately, but by combining several principles, the judgment will be simplified. For example, under the assumption a diagram satisfies Principles 1-10, Principle 13 is equivalent to the following principle:

Principle 13': For any $v \in V_5$ and the set $S = \{w \mid (V_4[w], v) \in E_4\}$ there exist $w_1, w_2 \in V_3$ and $x \in V_2$ such that $(w_1, V_4[\min S]), (w_2, V_4[\max S]) \in E_3$ and $(x, w_1), (x, w_2) \in E_2$ holds.

By using this property, we have a bit efficient way of verifying Principle 13. That is, for each node v in V_5, you have only to find the minimum and maximum ordinal numbers p and q where v is the rightward parent of $V_4[p]$ and $V_4[q]$ to determine whether the leftward parent's parent of $V_4[p]$ is the same as that of $V_4[q]$.

4. Support System for Drawing Clamshell Diagram

We have been developing a support system for drawing Clamshell Diagrams on the Web. The system works as a Web application. The server accesses to a database management system, which stores the nodes on the diagram together with the linkage and those outside the diagram (called temporary nodes). Another remarkable feature of the system is how to express the connection of nodes; the whole diagram is displayed as an aggregation of boxes. It has six columns but no wire connection. Instead the horizontal adjacency means the connection. Note that the vertical adjacency does not matter. Box representation is adopted to render the diagram using HTML effectively.

Figure 4 shows a screenshot of an ACD about what to learn in the introductory education of computer department. The interfaces for temporary nodes are hidden, below the table in reality. This chart was derived from the curriculum of our department, through the discussion and modification among the authors, some of whom teach the skills and the others have learned. The benefit of the Web application of our support system is that the users can collaborate with each other to complete a diagram; after one finishes a part of it, another person can take action to visualize a different part or to (re)combine the nodes. By resolving the problem of communication media of intention and exclusive access, our system will be a more practical multiplayer application.

5. Application to Source Code Reading

5.1. Representation using Clamshell Diagram

In this section we attempt to chart source codes by means of Symmetric Clamshell Diagram. The intended programs are limited to Open Source Softwares, since the codes and comments are expected to be written carefully and legibly with due consideration to the users other than the developers. Note that our approach belong to the static code analysis. We consider that generating source code for a diagram is attracting but infeasible.

Here we will explain the outline for making an SCD from a package of source file. After a given archive file (often called a tarball) is expanded, there exists a directory which contain all the source files, document files and others. The first step of viewing

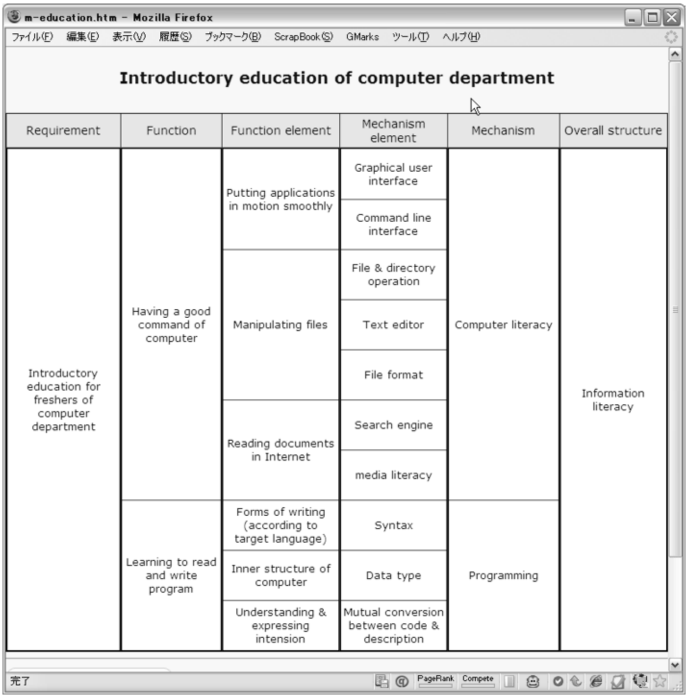


Figure 4. Screenshot of support system

SCDs is to form a right part tree structure where the name of that directory is labeled on the rightmost root. The level of the right part tree is unrestricted. Usually, it forms a hierarchy structure of root directory name, other directory name, file name, function, and procedure in descending order. Functions and procedures are derived from the concrete source code while the rest can be find from file name hierarchy. Those contents cover the right part of SCD. After then, the left part tree is constructed so that the whole graph may be symmetric and each node in the left part may be labeled as the description of file or directory, or as the comment on function or procedure corresponding to that in the right part. The structure (Fig. 5) is not a Clamshell Diagram discussed in Sect. 3 since it violates Principle 1, but by changing the level any other principles largely hold. Next step is to restrict the field of view in order to present an SCD. The origin which is to be fixed is a node in the right part tree, other than a leaf or a parent of a leaf. After fixing the origin, all you have to do is to gather the nodes which are less than 3 level downward (leftward) far from the selected origin together with those corresponding in the left part tree. When two subtrees are combined with each leaf connected, you can take a view of an SCD. In Fig. 5, for example, we can make up SCDs by the nodes of thick lines as well as the starred nodes. By switching over the origin, we will browse the whole source file in the style of SCD.

5.2. *Example: GNU Hello*

Figure 6 shows an example of SCD for GNU Hello, version 2.3[9]. Parenthetic phrases are specified manually and the other labels are extracted automatically from the file `hello.c`, which has and holds 193 lines with 18 comments of varying size.

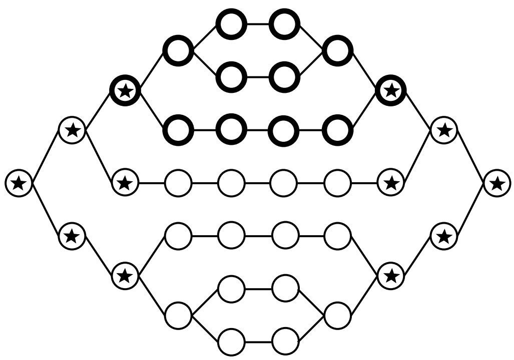


Figure 5. Diagram of a source file for SCD indication

<table>
<tr><th>Description of file</th><th>Description of function</th><th>Comment on procedure</th><th>Procedure</th><th>Function</th><th>File</th></tr>
<tr><td rowspan="15">print a greeting message and exit.</td><td>(Other than function)</td><td>String containing name the program is called with.</td><td>l.24: const char *program_name;</td><td>(Global declaration)</td><td rowspan="15">hello.c</td></tr>
<tr><td rowspan="7">(Main function)</td><td>Set locale via LC_ALL.</td><td>l.49: setlocale (LC_ALL, "");</td><td rowspan="7">ll.39-127: int main (int argc, char *argv[])...</td></tr>
<tr><td>Set the text message domain.</td><td>ll.53-54: bindtextdomain (PACKAGE, LOCALEDIR);...</td></tr>
<tr><td>Even exiting has subtleties.</td><td>l.62: atexit (close_stdout);</td></tr>
<tr><td>One goal here is having --help and --version exit immediately,...</td><td>ll.64-89: while ((optc = getopt_long (argc, argv, "g:hntv", longopts, NULL)) != -1)...</td></tr>
<tr><td>Print error message and exit.</td><td>ll.91-100: if (lose || optind < argc)...</td></tr>
<tr><td>Print greeting message and exit.</td><td>ll.103-104: if (t)...</td></tr>
<tr><td>TRANSLATORS: Use box drawing characters or other fancy stuff</td><td>ll.106-117: else if (n)...</td></tr>
<tr><td rowspan="5">Print help info.</td><td>TRANSLATORS: --help output 1 (synopsis)...</td><td>ll.140-141: printf (_("\...</td><td rowspan="5">ll.135-171: static void print_help (void)...</td></tr>
<tr><td>TRANSLATORS: --help output 2 (brief description)...</td><td>ll.145-146: fputs (_("\...</td></tr>
<tr><td>TRANSLATORS: --help output 3: options 1/2...</td><td>ll.151-153: fputs (_("\...</td></tr>
<tr><td>TRANSLATORS: --help output 4: options 2/2...</td><td>ll.158-161: fputs (_("\...</td></tr>
<tr><td>TRANSLATORS: --help output 5 (end)...</td><td>ll.169-170: printf (_("\...</td></tr>
<tr><td rowspan="2">Print version and copyright information.</td><td>xgettext: no-wrap</td><td>l.182: puts ("");</td><td rowspan="2">ll.177-193: static void print_version (void)...</td></tr>
<tr><td>t is important to separate the year from the rest of the message,...</td><td>ll.187-192: printf (_("\...</td></tr>
</table>

Figure 6. Clamshell Diagram of hello.c

A basic corresponding strategy upon procedure pairs a comment and the following code before a blank line. Exceptionally, if a comment is described just after control statement with opened brace, then the comment is associated with all the control block.

We have to point out that the automatic detection is somewhat less than perfect. For example, the comment "Print greeting message and exit" is associated with the procedure on ll.103–104, but the thing is that the comment should correspond to ll.103–126, which includes the output of a greeting message other than the well-known "Hello, world!" and

the exit call. Although it is not hard to bracket the “if ... else if ... if” clause on ll.103–124 by automatic processing, the function call of exit on l.126 just after a blank line would get away. That indicates the limitation to the way of splitting the codes by means of blank lines, rather than the defect of the coding standard.

The notion of corresponding relationship mentioned in Sect. 2.1 is also applicable to the diagrams for source code reading. An incomplete, unwise source file often possesses the code with no appropriate comment around it. Alternatively, through bug fixes and refactoring, it is not uncommon to see the code revised frequently while the collateral comment hold steady, to find the mismatch between the ideal and the real. Our approach helps the programmers or the reviewers not only to find the deficiency or the mismatch but to yield a clue of refinement of the intention as well as the codes in a large sense by observing a Clamshell Diagram multidirectionally, which has more to gain than a traditional way of review where one merely compares the present code with the comment.

For other programming languages, when a source file is described in a manner consistent with some coding standard in regards of codes and comments, we would easily draw up a set of rules to automatically extract the nodes for an SCD. The level of double tree constructed in advance depends on the programming language and the coding standard. For example, if a Java source file defines more than one class where there might be nested classes, then the layer of class should be placed between those of file and method, and we will find the column of class in a Clamshell Diagram.

6. Conclusions

In this paper, we have given a formal definition of Symmetric and Asymmetric Clamshell Diagrams by means of principles. Moreover a support system for drawing the diagram and an application to source code reading is reported. The Clamshell Diagram is an powerful, visual scheme which possesses both whole-part relationship and ends-means relationship. We hope that the diagrams will widespread for creating novel ideas and for analyzing some subject matter closely.

References

[1] S.J. Mellor and M.J. Balcer: *Executable UML: A Foundation for Model-Driven Architecture*, Addison-Wesley, 2002.

[2] H. Mase, H. Kinukawa, H. Morii, M. Nakao and Y. Hatamura: *Mechanical Design Support System Based on Thinking Process Development Diagram*, Transactions of the Japanese Society for Artificial Intelligence, Vol.17, pp.94–103, 2002 (in Japanese).

[3] J. Kawakita: *KJ method: A Scientific Approach to Problem Solving*, Kawakita Research Institute, 1975.

[4] K. Yagishita, J. Munemori and M. Sudo: *A Proposal and an Application of an Evaluation Method for Sentences of B Type KJ Method Based on Contents and Structures*, Transactions of Information Processing Society of Japan, Vol.39, No.7, pp.2029–2042, 1998 (in Japanese).

[5] T. Buzan and B. Buzan: *The Mind Map Book : How to Use Radiant Thinking to Maximize Your Brain’s Untapped Potential*, Dutton, 1994.

[6] R.B. Grady: *Successfully applying software metrics*, Computer, Vol.27, No.9, pp.18–25, 1994.

[7] G. Altshuller, L. Shulyak and S. Rodman: *40 Principles: Triz Keys to Technical Innovation*, Technical Innovation Center, 1998.

[8] M. Schlueter: *TRIZ for Perl-Programming*, Proceedings of TRIZCON2001, 2001.

[9] GNU Hello, `http://www.gnu.org/software/hello/`

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-484

Personalized Teaching of a Programming language over the web: Stereotypes and rule-based mechanisms

Konstantina Chrysafiadi and Maria Virvou
Department of Informatics, University of Piraeus

Abstract. In this paper we describe the student model that has been developed for a web-based educational application that teaches the programming language Pascal. The tutoring system is called Web_Tutor_Pas. The application adapts its responses to each individual student dynamically. The system's student model is web-based on a three-dimensional stereotype approach. One dimension concerns the knowledge level of the student the second dimension concerns the type of programming errors (logical or syntax) and the third concerns previous knowledge of the student on other programming languages. The system takes into account information about the student from the student model and using a rule-based mechanism, provides individualized instruction, examination and advice.

Keywords. Stereotypes, student model, teaching programming, personalisation

Introduction

Over the last decade, the Internet is used for providing several services to people all over the world. It is used for communication, searching for information, advertising, e-banking, e-commerce, e-government and e-learning. Indeed, in the field of education its use has been continually growing. Web-based educational applications offer easy access and also are independent of platform. Due to the absence of teacher and to the different characteristics of students, the web-based application has to adapt to the needs of each learner. Adaptive web applications have the ability to deal with different users' needs for enhancing usability and comprehension and for dealing with large repositories [1]. Adaptivity requires dynamic adaptation of the web application tailored to individual users' needs.

In the case of e-learning, adaptivity is achieved using the technology of Intelligent Tutoring Systems (ITS). More specifically, the student modeling component of an ITS is mainly responsible for individualization of an educational application to the particular needs of a student. For providing effective individualized tutoring it is essential to construct both short term and long term student models [2][3]. The short term student model is responsible for checking and explaining a learner's errors each time s/he interacts with the system. The long term student model is responsible for monitoring a learner's actions throughout many interaction sessions. This is achieved by recording a user's history and making inferences about her/his progress.

In this paper we describe the student modeling component of a web-based educational application that teaches the programming language Pascal. We have selected this knowledge domain because there is a significant scientific interest in the effective tutoring of programming courses [4][5][6][7]. Our application provides personalized tutoring to each learner. For this purpose the system performs student modeling which is based on stereotypes. Stereotyping has been used extensively in many information systems to represent user groups and/or to generate individual user models [8][9][10]. The user model of our application has three dimensions. One dimension concerns the knowledge level of the student. The other dimension concerns the type of programming errors that is if an error concerns the syntax of a command or if it is logical. The third dimension concerns previous knowledge of the student on other programming languages. The student model of Web_Tutor_Pas is responsible for diagnosing possible errors of students while they are examined through a dynamic test, tracing students' progress and presenting in a suitable way the domain knowledge taking into account the learner's needs. Adaptivity is achieved using rule-based mechanisms.

1. The Domain Knowledge

One of the most important components of an adaptive educational application is the representation of knowledge. To enable communication between system and learner at content level, the domain model of the system has to be adequate with respect to inferences and relations of domain entities with the mental domain of a human expert [11]. Taking this into account, the domain knowledge of our application is organized into an hierarchical tree, with domain topics as intermediate nodes and learning objectives as leaf nodes [12]. The domain topics consist of course modules. The full collection course modules cover the whole programming domain to be taught. There are topics which concern declarations of variables and constants, expressions and operands, input and output expressions, the sequential execution of a program, the selection statement (if statement), the iteration statements (for loop, while loop and do – until loop), sorting and searching algorithms, arrays, functions and procedures. Each topic has its learning objectives that are classified according to Bloom's taxonomy [13]. Learning objectives determine the concepts that must be understood and learned in each chapter. For example, the learning objectives for the topic of variables are:

- To learn the types of variables and understand their differences and how to use them
- To learn how to declare variables
- To learn how to select the appropriate type of variable each time
- To learn the rules about variables' names
- To learn to use variables saving memory

The hierarchy of the tree depicts the difficulty level of the domain topics and the order in which each topic has to be taught. The creation of the hierarchy is based on the knowledge domain, on the logic of programming languages and on the dependencies that exist among the components of a programming language. For example, the teaching of variables and operands proceeds to the teaching of the if statement and the learning of a sorting algorithm presupposes the existence of knowledge of selection and iteration statements. The domain knowledge also encodes relations between different chapters and the categorization of these chapters according to their difficulty level.

2. Student Modeling

It has long been recognized that in order to build a good interactive computer system with an heterogeneous user community, it is necessary the development of individual user models [14]. The student modeling component is responsible for dynamically representing the emerging knowledge and skills of the student [15] and for inferring the learner's knowledge and misconceptions for her/ his behavior [16].

The user model of our web-based educational application is based on stereotypes. According to Kay [17] the stereotype has a set of trigger conditions, {tMi}, where each {tMi} is a boolean expression based upon components of the student model, and a set of retraction conditions, {rMi}. The primary action of a stereotype is illustrated by Eq (1) and a stereotype is deactivated when any of the retraction conditions becomes true: Eq (2). In our user model a set of triggering and retraction conditions represents each learner's cognitive state with respect to her/his progress or non-progress, to the kind of programming errors that s/he does and her/his previous knowledge on programming, while s/he interacts with the system. More specifically, our user model is based on a three-dimensional stereotype approach. One dimension concerns the knowledge level of the learner, the second dimension concerns the type of programming errors, if an error is syntactic or logical, and the third dimension concerns a learner's previous knowledge on programming languages.

$$if\ \exists i,\ tM_i = true \rightarrow active(Mg) \quad \textbf{(1)}$$

$$\exists j,\ rM_j = true \rightarrow not\ \ activate(M) \quad \textbf{(2)}$$

For the definition of the first dimension of our model we advised the conceptual framework for analyzing students' knowledge of programming that was developed by McGill and Volet [18] and the evaluation method of knowledge of programming that was developed by deRaadt [19]. McGill and Violet discern three knowledge types in the view of cognitive psychology: declarative (the basic knowledge of an object), procedural (how to use declarative knowledge for problem solving and decision making), strategic (upper knowledge level), and three knowledge types in the view of educational research: syntactic (basic knowledge), conceptual (be able to combine knowledge, analytical thought) and strategic (integrated knowledge). De Raadt suggests five knowledge levels:

- No answer: learner has no knowledge of programming and cannot give an answer
- Pre-structural: substantial lack of knowledge of programming structures
- One-structural: learner is able to describe a part of code
- Multi-structural: learner is able to describe a program line to line
- Relational: learner is able toy describe the whole of a program.

The first dimension consists of eight stereotypes (table 1). The stereotype categories range from "novice" users, who do not have a structural knowledge of programming and are unable to give an acceptable answer in most cases, to "expert" users, who are able to select, use and combine programming structures creating complex programs. A learner is classified to a stereotype category according to which chapters the user knows and how well. Stereotype 1 includes the novice users, stereotype 2 includes the learners that know the basic concepts of programming

(variables, constants, operands, expressions, input – output expressions, sequential execution) and the next stereotype includes the users that know the selection statement (if, if-else if, nested if). Learners who know how to build programs using the for statement belong to stereotype 4, since the learners who know how to use the statement while and do-until, belong to stereotype 5. Stereotype 6 includes the users that know arrays, stereotype 7 includes the advanced users who know the most significant concepts of programming and can build complex programs such as sorting and searching algorithms, and finally stereotype 8 includes the expert users who know also the process of subprogramming.

Table 1. Stereotypes of first dimension

Stereotype	Knowledge type	Knowledge level according to de Raadt
Stereotype 1	no knowledge	Level 1
Stereotype 2	declarative – syntactic	Pre-structural
Stereotype 3	declarative – conceptual	One-structural
Stereotype 4	procedural – syntactic	One-structural
Stereotype 5	procedural – syntactic	Multi-structural
Stereotype 6	procedural – conceptual	Multi-structural
Stereotype 7	procedural – conceptual	Relational
Stereotype 8	strategic	Relational

The second dimension consists of two stereotypes, namely prone to syntax errors and prone to logical errors. Syntax errors are recognized if they belong in one of the following categories:

- Anagrammatism of commands' names
- Omission of the definition of data
- Using invalid command names e.t.c..

They, usually, indicate that the learner has not read carefully and has not known adequately the chapters that correspond to her /his knowledge level. Logical errors are usually errors of design and occur in case of misconceptions of the program and of the semantics and operation of the commands. They, usually, indicate that the learner has a difficulty in understanding the instructions and their logic. Indeed, when a learner makes more logical and trivial syntax errors, then it means that she/he has read the corresponding chapters but she/he has difficulty in understanding instructions' function. Thus, the domain knowledge has to be presented in a different suitable way being adaptive to the learner's needs. More specifically, in this case, the presentation of the domain knowledge has to include more examples and diagrams.

The third dimension consists of stereotypes that are associated with other programming languages that the learner may already know. These stereotypes are restricted to programming language C.

3. Adaptivity Based on Student Model and Rules

Web_Tutor_Pas is accessed by many users with different characteristics, needs and knowledge level and in situations where the human teacher is absent. Therefore, it has been developed in such a way that it can respond to each learner's needs. The adaptivity is achieved obtaining information about the learner from her/his student model and using a rule-based mechanism.

When a user interacts with the system for the first time, it is assumed that she/ he is a novice user. At the next interactions with the web-based educational application the

stereotype category of the user can change according to her/ his performance determined by a test that she/ he has to complete at the end of each instructional process. This test includes four types of exercises. It includes true/false exercises, multiple choice exercises, fill in the blank space exercises, where the student fills in a symbol or a command in order to complete a program and exercises in which users have to put certain parts of a program in the right order. Each exercise is associated with two categories of errors. The first category corresponds to one or more learning objectives and represents the difficulty and the knowledge level. More specifically, there are thirty one error subcategories which are associated with the domain knowledge and the percentage of these errors that a student made while s/he was completing the system's test, determine the first dimension stereotype of him/her (table 2).

Table 2. Error categories and knowledge level

Error categories	Stereotype	General domain topic
1. constants & variables	Stereotype 1: "novice learner"	Basic concepts of programming
2.a simple program's structure		
3. input-output statements		
4. assignment statement		
5. arithmetic operators		
6. mathematic functions		
7. comparative operators		
8. logical operators		
9. if statement	Stereotype 2	The if statement
10. nested if statement		
11. finding max, min		
12. for statement	Stereotype 3	The for statement
13. finding sum in a for statement		
14. finding average in a for statement		
15. counting in a for statement		
16. finding max, min in a for statement		
…….	…….	…….
30. procedures	Stereotype 8: "expert learner"	Subprogramming
31. functions		

The second error category corresponds to the kind of programming error (syntax or logical) and represents either the lack of knowledge concerning programming rules or the degree of carelessness of a learner or the ability of a learner to understand the logic of programming. More specifically, a syntax error, usually, indicates that a learner has not read the corresponding domain knowledge, or s/he has read it but s/he has not learn it, or s/he is careless. On the other hand, a logical error indicates that a learner has a difficulty in understanding the function and logic of instructions and programming. The system checks the percentage of syntax and logical errors that a student made in each topic of the knowledge domain and gives an explanation about them according to table 3.

Table 3. Explanation of errors

Syntactic	Logical	State
60%-100%	X	The learner has not read the corresponding topic
20%-60%	X	The learner has read the corresponding topic but s/he has not learned it
0-20%	80%-100%	The learner has read the corresponding topic but s/he has a difficulty in understanding it

For example, two students used the system. In the first interaction with the system they are classified as novice users (stereotype 1) and completed a test that involved exercises which concerned the basic concepts of Pascal. The results are depicted in figure 1. We notice that the student A made less than twenty percentage errors in each error category of his stereotype, so s/he transits to stereotype 2. On the other hand, the student B made 29% errors on variables and constants, 33% errors on the assignment statement and 50% errors on the arithmetic operators. Therefore, s/he does not transited to the stereotype 2 and Web_Tutor_Pas advises the student about the topics that s/he should read again (figure 2). At the next interaction with the application, the system adviss students' models and presents the suitable theory part (figure 3). Furthermore, Web_Tutor_Pas uses different icons and font types to inform the student of how well s/he knows each topic and whether s/he has to study again a concept or not [20].

Results of student A:

Variables & Constants	A simple program's structure	Input-Output statements	Assignment statement	Arithmetic Operators & Precedence	Mathematic Functions	Comparative Operators	Logical Operators	TOTAL SYNTAX ERROS	TOTAL LOGICAL ERRORS
18% SYNTAX:100% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	17% SYNTAX:0% LOGICAL:100%	0% SYNTAX:0% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	67%	33%

Results of student B:

Variables & Constants	A simple program's structure	Input-Output statements	Assignment statement	Arithmetic Operators & Precedence	Mathematic Functions	Comparative Operators	Logical Operators	TOTAL SYNTAX ERROS	TOTAL LOGICAL ERRORS
29% SYNTAX:100% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	33% SYNTAX:50% LOGICAL:50%	50% SYNTAX:100% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	0% SYNTAX:0% LOGICAL:0%	20% SYNTAX:0% LOGICAL:100%	71%	29%

Figure 1. Test's results

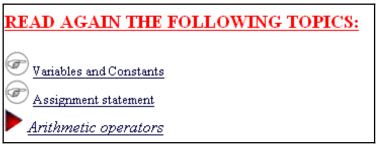


Figure 2. List of chapters for reading according to advice generator (: better study, :read again some points)

A significant point, concerning the operation of the system, is the fact that in the intermediate, advanced and expert levels of knowledge the tests become more complex. More specifically, the exercises are more difficult and combine elements of all the chapters that a user has been taught. For example, an exercise with a sorting algorithm requires knowledge on variable declaration, selection and iteration statements and arrays. Therefore, through the test the system can infer whether the learner has forgotten a topic or if she / he has not learned it or if she/ he has assimilated it. Once the system infers that the learner has forgotten something from the previous chapters or she / he has not assimilated a topic, it responds directly to this situation by classifying the learner to a previous appropriate stereotype. Indeed, if a learner is attached to an advanced stereotype and makes errors that correspond to previous knowledge level and these errors are made due to the learner's knowledge on another programming language (for example s/he knows C and writes an assignment statement with =, rather than :=), the system does not classify her/him to a previous stereotype, but it points out the error.

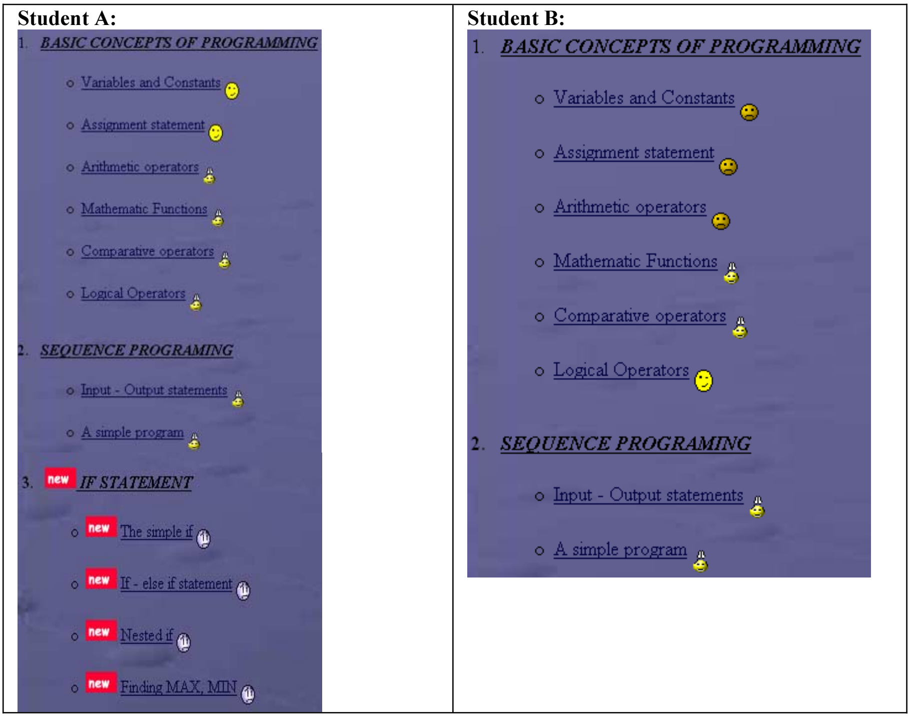


Figure 3: Table of course contents

For example, student A and student B are assigned to stereotype 2. After having completed the test which involved exercises on the if statement, the system discovers that both made more than 20% errors on the assignment statement which corresponds to stereotype 1, and more specifically they used symbol = rather than symbol :=. Then, Web_Tutor_Pas advices the third dimension of our system's stereotypes and is informed that student A already knows the programming language C, but student B does not know another programming language. Thus, it is assumed that Student A used the symbol = for the assignment knowledge due to his/her previous knowledge on C, it stresses the error but it does not classify her/him to stereotype 1. On the other hand, the system concerns that student B made errors on the assignment statement due to lack of knowledge on this topic. So, student B transits to stereotype 1.

In view of above, we can say that the system, each time, has to make a decision about the stereotype that the learner has to be assigned to the appropriate stereotype which depicts his/her knowledge level and abilities. At each interaction of a learner with Web_Tutor_Pas, the system has to decide which stereotypes have to be activated and which stereotypes have to be deactivated. It is achieved using the information of the student model and following a rule-based technique (figure 4).

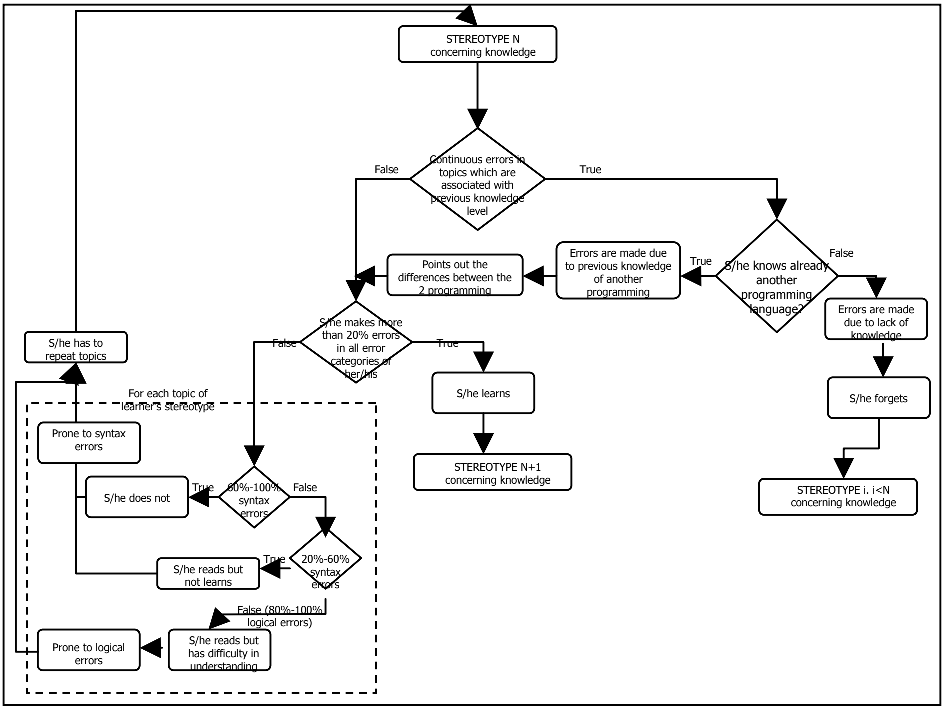


Figure 4: System's operation to make a decision about the learner's stereotype (triggering and retraction)

4. Conclusion

Our target in this paper was to create a web-based educational application which teaches the programming language Pascal, providing individualized support. The personalized support is realized due to the application's user model which is based on a three-dimensional approach. In particular, the system uses stereotypes which are determined depending on the learner's knowledge level at programming, on the types of programming errors (syntactic or logical) that a learner can make and on the knowledge of learner on other programming languages. Our system uses this user model and follows a rule-based technique to provide adaptive instruction. In particular, the system provides individualized instruction, individualized support during the navigation through course material, individualized test and personalized advice.

References

[1] S. Garlatti and S. Iksal, "Declarative Specifications for Adaptive Hypermedia Based on a Semantic Web Approach", *User Modeling*, 2003, p.p. 81-85

[2] E. Rich, "User Modelling via Stereotypes", *Cognitive Science*, 3 (4), 1979, pp. 329-354.

[3] E. Rich, "Users are individuals: individualising user models", *Int. J. Human-Computer Studies*, 51, 1999, pp. 323-338.

[4] P. Byckling and J. Sajaniemi, "A role-based analysis model for the evaluation of novices' programming knowledge development", In Proceedings of the 2006 international workshop on Computing education research, Canterbury, United Kingdom, 2006, pp. 85-96

[5] P. Grunbacher, N. Seyff, R. Briggs, Hoh Peter In, H. Kitapsi and D. Port, "Making every student a winner: The WinWin approach in software engineering education", Journal of Systems and Software, 80 (8), 2007, pp. 1191-1200

[6] F. Wei, S.H. Moritz, S. M. Parvez and G. D. Blank, "A student model for object-oriented design and programming", Journal of Computing Sciences in Colleges, 20(5), 2005, pp. 260-273

[7] A. Woszczynski, H. M. Haddad and A.F. Zgambo, "Towards a model of student success in programming cources", In Proceedings of the 43rd annual Southeast regional conference, Kennesaw, Georgia, 2005, pp. 301-302

[8] T. Clemmensen, "Four approaches to user modelling – a qualitative research interview study of HCI professionals' practice", *Interacting with computers*, 16 (4), 2004, pp. 799-829

[9] XM. Zhang and H. Han, "An empirical testing of user stereotypes of informational retrieval systems", *Information Processing & Management,* 41 (3), 2005, pp. 651-664

[10] LN. Michaud and KF. McCoy, "Empirical derivation of a sequence of user stereotypes for language learning", *User modeling and user-adapted interaction,* 14 (4), 2004, pp. 317-350

[11] C. Peylo, W. Teiken, C. Rollinger and H. Gust, "An ontology as domain model in a web-based educational system for prolog", *In Proceedings of the 13th International Florida Artificial Intelligence Research Society Conference*, eds. J. Etheredge and B. Manaris, AAAI Press, Menlo Park CA, 2000, pp. 55-59.

[12] A. Kumar, "Rule-Based Adaptive Problem Generation in Programming Tutors and its Evaluation*", In Proceedings of the 12th International Conference on Artificial Intelligence in Education,* Amsterdam, 2005, pp.35-43.

[13] B.S. Bloom, "Taxonomy of Educational Objectives", Handbook I: The Cognitive Domain. New York: David McKay Co Inc., 1956

[14] P. Brusilovsky, "Methods and techniques of adaptive hypermedia", *User Modelling and User-Adapted Interaction*, 6, 1996, pp. 87-129

[15] D. Chin D, "Empirical evaluation of user models and user adapted systems", *User Modeling and User-Adapted Interaction,* 11, 2001, pp. 181-194.

[16] H. Nwana, "User Modelling and User Adapted Interaction in an Intelligent Tutoring System", *User Modeling and User-Adapted Interaction*, 1(1), 1991, pp. 1-32.

[17] J. Kay, "Stereotypes, Student Models and Scrutability*", Intelligent Tutoring Systems*, 2000, pp. 19-30.

[18] T.J. McGill and S.E. Volet, "A Conceptual Framework for Analyzing Students' Knowledge of Programming", *Journal of Research on Computing in Education,* 29, 1997, pp. 276-297.

[19] M. de Raadt, "A Review of Australasian Investigations into Problem Solving and the Novice Programmer", *Computer Science Education,* 17(3), 2007, pp. 201-213.

[20] V. Tsiriga and Virvou M., "Evaluating the intelligent features of a web-based intelligent computer assisted language learning system", *International journal on artificial intelligence tools*, 13(2), 2004, pp. 411-425

Software Engineering Methods for Intelligent Tutoring Systems

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-495

Requirements Engineering Education for Professional Engineers

Takako NAKATANI [1]
GSBS, Univ. of Tsukuba

Abstract. We have developed an educational course for professional engineers to improve their abilities in requirements engineering, especially, the ability to fully grasp problems of the real world in order to elicit requirements. This is essential for engineers to establish valid software requirements specifications (SRS). In order to improve their abilities, we teach multiple methods on RE: e.g. SSM, problem flames and goal-oriented analysis methods, as well as object-oriented modeling methods with UML. In this article, we outline the course and the results of our experiences. The course is evaluated qualitatively.

Keywords. Requirements Engineering, Education

1. Introduction

The importance of requirements engineering (RE) has been emphasized since the Boehm reports[6]. There are several countermeasures to improve our Software Requirements Specifications (SRS). One of the ways is to define the quality characteristics of the items in the developing system. ISO/IEC9126[17] provides us the quality characteristics: i.e. functionality, reliability, usability, efficiency, maintainability, and portability. Fulfilling the recommendation of the IEEE std. 830-1998[16] is another way. It provides a good guide for requirements engineers to validate the quality of the contents of his/her SRSs. It also recommends us to express the rationale and the background of requirements in the "Assumptions and dependencies" section.

Our main problem is that most real SRSs lack the rationale or background descriptions of requirements. There is no standardized way to write the rationale of the requirements. Here is an amusing anecdote told by a project manager.

> There are engineers who can produce thick files of SRSs from the rich information on stakeholders' real requirements. They can also produce the same volume of specifications from a small amount of stakeholders' requirements.

This means that these engineers are in a culture that can form SRSs independently of stakeholders' needs. The culture is worse yet is that no reviewer can distinguish the later from the former. Unfortunately, the strongest evidence shows that, unsuccessful requirements elicitation is the main cause for delaying projects, and thus, is not enough

[1]Graduate School of Business Sciences, University of Tsukuba; 3-29-1, Otsuka, Bunkyo, Tokyo, Japan. Tel.: +81-3-3942-6869; Fax.: +81-3-3942-6829; E-mail: nakatani@gssm.otsuka.tsukuba.ac.jp

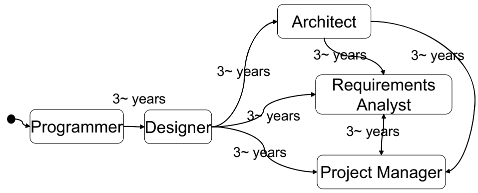


Figure 1. Japanese engineers' careers pass in general.

to overcome the cultural obstacles in place. Our course aims at motivating engineers to define the context of the developing system. A context would provide the rationale and evidence of the requirements adequately for the readers of SRSs.

Figure 1 represents the ordinal Japanese software engineers' careers pass. They begin an analyst life after approximately ten years of development experiences. In this paper, we refer to such an analyst as a professional engineer. Consequently, we can assume that professional engineers have programming skills, designing skills, and communication skills, as well as team working skills. This assumption most likely does not apply to undergraduate students. Teaching these skills is still our challenge. However, that is out of the scope of this paper.

Our educational course is designed to motivate trainees to learn RE in order to write SRSs with the rationales and evidence of the real world in which the requirements come from. Therefore, the course is initiated from a practice of understanding the problematic real world, rather than from eliciting requirements. To understanding the real world, it is essential to integrate business analysis and requirements engineering[15]. In the course, we use RODAN with a metamodel that includes metamodels of multiple methods[22,23]. RODAN is an integrated RE method[7]. It is composed of SSM (Soft Systems Methodology)[10], KAOS[9], Use cases[18,1], and Eriksson&Penker's business modeling method[14]. Beus-Dukic et al. mentioned that teaching use case modeling is not straight forward[4]. We agree with them. However, professional engineers study technology by themselves, if and only if they believe it as an essential technology. Our main aim is to motivate engineers to start learning RE. Thus, we have conducted a workshop style course[27] in which the trainees can face a lot of problems, and can also evaluate the continuous process of trial and error.

Goal-oriented analysis methods[2,12,26] emphasize the importance of capturing the "why" aspect of requirements. If an engineer knows a goal-oriented analysis method, he/she can build a goal model. The goal model seems to represent the rationale of requirements, then, how can we evaluate its adequacy?

If we have the following evidential information, we can validate SRSs.

1. (from a personal perspective) Who needs the requirements and why?
2. (from an organizational perspective) Why the requirements should be highly prioritized?
3. (from an analyst perspective) Why should analysts interview the stakeholders?

Our approach is to answer the above questions by applying RODAN.

The remaining part of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 describes an overview of the course. Section 4 presents the workshop assignment and its artifacts as an example. In Section 5, we present our ob-

servations and the course evaluation. In the final section, we draw our final conclusions from the study.

2. Related works

The basic RE method, BaRE, was designed to cover all the key issues in RE in order to introduce the knowledge of how to do RE[24]. There is currently an educational course developed for undergraduate students in order to provide the students with specific abilities/skills to face the demands coming from the professional software engineering market.[19]. The 16 hours course is composed of modules: i.e. software quality, requirements documentation, a requirements elicitation workshop, techniques and methods for quality analysis, and tools.

There is also a game style educational course. Re-O-Poly is a game designed to educate requirements engineering methods and interviews[25]. The players meet various situations that RE engineers sometimes confront. Thus, they can learn the ways and the time required to apply RE methods by playing the game. We think that the trainees should learn the mind of their stakeholders. In this meanings, role playing exercise seems to be adequate for learning RE[27].

Berenbach and et al. have introduced a unified requirements model that integrates features, use cases, as well as hazard analysis[3]. As the scope of RE expands, multiple methods should be applied in an empirical situation. RODAN is a method for RE by integrating multiple methods.

3. Course plan

We have developed a two and a half day workshop[21]. Each team, consisting of four to five members, is required to present an SRS. The SRS is evaluated by the members of the other teams, who play the role of stakeholders. This allows every team member the chance of playing both a developer role and a stakeholder role.

3.1. Course overview

Requirements elicitation is an activity for the stakeholders benefit. We must consider the fact that stakeholders' needs have originated from their world views. Therefore, the course is initiated by focusing on personal viewpoints.

Figure 2 shows the activities of each educational step along with their exit criteria. The course consists of three periods.

- The first period (eight hours):
  This period aims at learning the relationships between the stakeholders' viewpoints and their perspectives. After the first period, the trainees will be able to reflect the problematic real world and, answer the first question: "Who needs the requirements and why?" with evidential information.
- The second period (six hours):
  The second period aims at extracting organizational goals and requirements. After the second period, the trainees will be able to answer the questions: "Why the

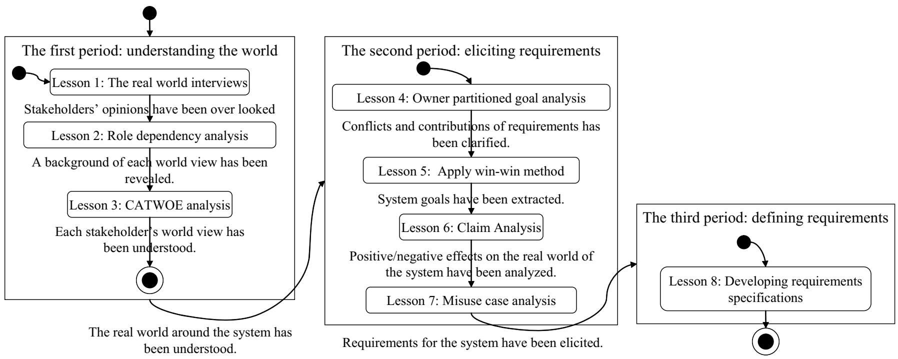


Figure 2. Course activities and their exit criteria.

requirements should be highly prioritized?" and "Why should we interview the multiple stakeholders directly?"

- The third period (four hours):
  The trainees develop an SRS by referring to the models of the workshop and review its adequacy with their stakeholders.

3.1.1. The first period

This period aims at learning the relationships between stakeholders' viewpoints and their perspectives. The trainees are expected to learn that the real world is observed by each person differently. Stakeholders usually view their world from their personal views and organizational views. Analysts must untie those mixed viewpoints and clarify substantial problems. In this period, we use rich pictures and CATWOE definitions introduced by SSM[10], as well as RDM of RODAN[22,23].

Rich pictures are useful to depict a stakeholders' situation or environment. The CATWOE[2], which is interpreted as, "the *actor* is ordered by the *owner* to *transform* the present problematic situation to the accomplishable future situation for their *customers* in the given *environment*, represents an owner's intention for the future world." The "W" represents the *Owner*'s viewpoint and his/her perspectives.

The course proceeds with a lecture, exercise, and evaluation cycle as follows:

- Lesson 1 (four hours):
  - ∗ Lecture: RE introduction and Rich picture.
  - ∗ Exercise: Draw rich pictures in order to look over the real world situation with stakeholders' opinions by interviewing with their stakeholders.
  - ∗ Evaluation: Review their pictures with their stakeholders.
- Lesson 2 (two hours):
  - ∗ Lecture: Role Dependency Model (RDM)[22], that extended Strategic Dependency Model of i*[26] by adding three kinds of roles: i.e. actor, owner, and customer. The RDM is interpreted as, "the owner orders the actor to do some-

[2]CATWOE is an abbreviation of *C*ustomer, *A*ctor, *T*ransformation process, *W*orld view, *O*wner, and *E*nvironment.

thing for the customers." We can understand the adequate goals that should be achieved in the stakeholders' organization by tracing the role dependencies.
  * Exercise: Role Dependency analysis.
  * Evaluation: Review their RDMs with their stakeholders.

- Lesson 3 (two hours):
  * Lecture: CATWOE.
  * Exercise: CATWOE analysis.
  * Evaluation: Review and validate the result of their analysis by listing the problems of the stakeholders' world.

3.1.2. The second period

This period aims to extract requirements and their alternatives to solve organizational problems. We use goal-oriented analysis methods, UML, use case model[18] with misuse cases[1], and scenario analysis[8]. Goal-oriented analysis methods including KAOS[9] is used to analyze the commonality and variability of requirements. The NFR framework[11] is used to analyze the positive and negative contributions of the subgoals in relation to achieving the various supergoals. Models with UML can be used to represent as-is and to-be situations from the structural perspective, dynamic perspective, and collaboration perspective. We apply Owner Partitioned Goal Model (OPGM) that extended the concept of the KAOS and NFR framework with partitions of the top goals for each owner. The personal perspective is represented in each partitioned lane in an OPGM. Models with UML can be used to represent as-is and to-be situations from the structural perspective, dynamic perspective, and collaboration perspective. The as-is models represent the real problematic situations. The to-be models represent the accomplishable future situations. Therefore, these models are used as evidence of CATWOE definitions.

- Lesson 4 (one hour):
  * Lecture: KAOS and OPGM
  * Exercise: OPG analysis
  * Evaluation: Review their OPGMs with their stakeholders.

- Lesson 5 (one hour):
  * Lecture: The way to elicit requirements from the OPGMs.
  * Exercise: Elicit the system requirements to fulfill win-win situations among the stakeholders.
  * Evaluation: Review the system requirements with their stakeholders.

- Lesson 6 (two hours):
  * Lecture: Erikkson & Penker's business modeling with UML, and scenario analysis[8] to define the valid requirements that do not threaten the positive stakeholders' requirements.
  * Exercise: As-is and To-be Analysis, as well as scenario analysis.
  * Evaluation: Review the adequacy of the requirements and whether it affects stakeholders positively or negatively.

- Lesson 7 (two hours):

  - ∗ Lecture: Misuse cases[1] and Abuse frames[20] to protect the system from negative actors.
  - ∗ Exercise: Apply Misuse case analysis.
  - ∗ Evaluation: Review the acceptability of the requirements and whether it can prevent the system from the negative stakeholders' intention.

3.1.3. The third period

We expect trainees to produce requirements specifications[16]. Each analyst group presents their SRS to their stakeholders.

- Lecture: IEEE std. 830-1998.
- Exercise: Develop an SRS from the models of the exercises.
- Evaluation: Review of SRS with the stakeholders in order to evaluate whether it understands the rationales clearly or not.

4. Course experience

4.1. Workshop assignment

The workshop was conducted with three to five groups. Each group was composed of four to five engineers with less than ten years software development experience. The following assignment was used:

> *Define a set of software requirements specifications for an effective meeting support system: thus, avoiding ineffective meetings.*

We used "Learned Memos (LM)", which are shared post-it note pads for listing up what they learned. The trainees collected LMs after each review session and put them up on the wall.

4.2. Artifacts

In the workshop, requirements elicitation started from interviews conducted through role playing. Stakeholders' opinions were represented in a rich picture. Then, the analyst team defined the role dependencies in the picture through the *actor*, *customer* and *owner*. Thus, the rich picture became RDM. The RDM example is depicted in Figure 4.

Figure 4 shows that there are various people in the meeting domain. We guided the trainees to grasp the stakeholders' world views. The following CATWOE[10] examples can also help them understand *how* to grasp the stakeholder's world views. The examples are defined for the same C, A, T and E, but with different owners. The definitions represent each W as being different for each O.

- *Customer*: A person who wants to call a 10 minute meeting.
- *Actor*: A person who arranges the meeting.
- *Transformation process*:
  from a situation: "a short term meeting cannot be held", transformed to the future situation: "a short term meeting can be held."
- *Environment*: Meeting rooms are available for short term meetings.

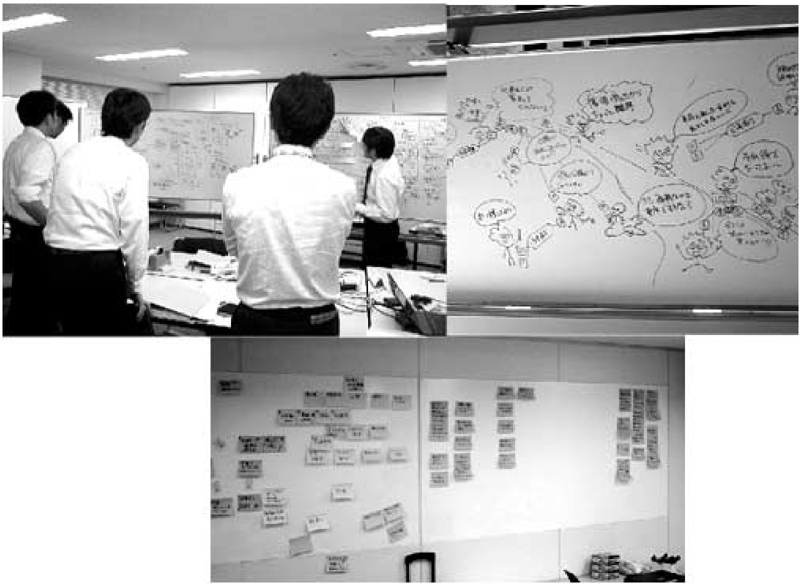


Figure 3. A scene of the course (A review, RDM and LMs)

- case1
  - ∗ *Owner*: A person who wants to call a 10 minute meeting.
  - ∗ *World view*: Short term meetings help us understand each other smoothly and, help make quick decisions that improve our business productivity.
- case2
  - ∗ *Owner*: A person who arranges the meeting.
  - ∗ *World view*: I'm not expecting my boss to assess me as an incapable person, because of the scheduling problem of meeting rooms.

The defined CATWOE can be interpreted in a structure composed according to why, what and how information. For example, the *W* is "why" for execution of *T*, and the execution of *T* is "what" for the *E*. Thus, the workshop was ready to begin goal oriented analysis. These three items could be the nodes of a goal tree.

The purpose of this training is to help the trainees recognize the extent of the solution domain. The goal model has several leaves for the root problems. Thus, the trainees grasp requirements based on the real remarks of their stakeholders. Requirements triage[13] and win-win negotiation[5] will be conducted in a real project.

5. Discussions

5.1. The strength and the weakness

As we mentioned in the first section, the course was developed for professional engineers. Our assumption of the trainees was that the trainees were well grounded in software

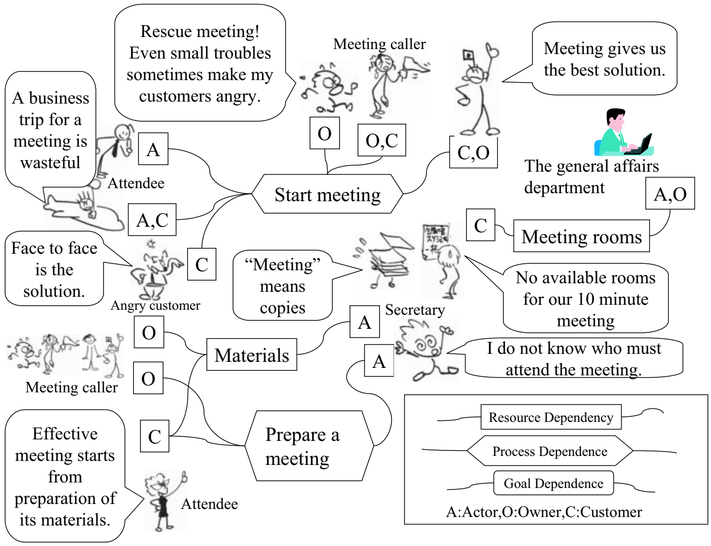


Figure 4. An example of the RDM.

engineering. We have conducted the course six times in the past two years. The first three courses did not succeed, because the assumption was wrong. A novice engineer who is not familiar with the static and dynamic modeling techniques needs an additional week in modeling training.

The trainees of our course are selected and ordered to attend the course by their bosses. The workshop role playing game was an effective exercise to make them join the course.

During the first period of the workshop, the purpose was set to learn how many different viewpoints exist in a domain. Since the trainees were experienced engineers, we could expect them to have faced requirements changes many times. Therefore, it was not difficult to teach them the necessity of analyzing stakeholders' world views.

5.2. *RE methods integration*

In this paper, we introduce several methods that are applied in the course. SSM is one of these methods, but it is not a mandatory one. Before we developed the course, we constructed a metamodel for stakeholders' needs or, requirements themselves[22]. The metamodel can accept other methods if their products fit the metamodel. Consequently, the trainees are responsible for selecting adequate methods and applying them as one of their requirements elicitation methods. Each software development company has their own standard development process, including requirements elicitation. The way to integrate RE methods in future projects is the responsibility of the trainees themselves. They may apply their familiar methods instead of SSM or OPGM

5.3. *Monitoring for evaluation*

We evaluate the course by using a simple questionnaire and asking for the trainees' candid comments. In a course with 15 trainees, we collected 67 and 70 LMs as comments

of a stakeholder role and an analyst role, respectively. Furthermore, there were 28 LMs as evaluation of RODAN, and 6 LMs for the SRS as an artifact of the exercise. The followings are the comments from the trainees.

- Unproductive questions:
When a trainee with a stakeholder role was asked by an analyst: "This is our understanding of your intention.": he could not reply "no", even though the model did not represent his world completely. He seemed to learn a question which confuses the stakeholder.
- Requirements were not for software engineers:
A trainee found that requirements were not the art of an engineer, but of the stakeholders. She also mentioned that they had to offer solutions that the stakeholders could accept, rather than those the engineers wanted to build.
- Whose rationale?:
A trainee commented that he understood the reason why our stakeholders changed their requirements frequently.
- A solution was not a solution for every one:
A trainee realized that there were different problems and solutions for every stakeholder. He mentioned that one solution could not solve everyone's problem.
- Requirements needs rationale:
There are several trainees who mentioned that "some of the requirements in the SRS did not have rationales. When the requirements were decomposed into detailed functional requirements, their rationales became vague." Thus, the course succeeded in helping the trainees focus on the rationale of the requirements.

We have a plan to produce a checklist from the LMs for the next course execution. Developing a method to evaluate the course quantitatively is our future work.

6. Conclusions

This paper introduced an RE training course for professional engineers. We can conclude that the course is constructed to make trainees answer three questions. A quantitative evaluation will be our future work.

Acknowledgement

We would like to thank all trainees, and we would also like to offer a special thanks to the training staff for helping to improve our course.

References

[1] ALEXANDER, I. Misuse cases use cases with hostile intent. In *IEEE Software* (January/Frbruary 2003), IEEE, pp. 58–66.
[2] ANTON, A. I. Goal-based requirements analysis. In *Proc. of the ICRE'96* (1996), IEEE, pp. 136–144.
[3] BERENBACH, B., AND WOLF, T. A unified requirements model; integrating features, use cases, requirements, requirements analysis and hazard analysis. In *Proc. of the International Conference on Global Software Engineering (ICGSE 2007)* (2007), IEEE, pp. 197–203.

[4] BEUS-DUKIC, L., AND MYERS, C. Use and abuse cases. In *1st International Workshop on Requirements Engineering Education and Training (REET'05)* (2005).

[5] BOEHM, B., BOSE, P., HOROWITZ, E., AND LEE, M. J. Software requirements negotiation and renegotiation aides: A theory-w based spiral approach. In *Proc. of the ICSE'95* (1995), ACM, pp. 243–253.

[6] BOEHM, B. W. *Software Engineering Economics*. Prentice Hall, 1981.

[7] BRINKKEMPER, S., SAEKI, M., AND HARMSEN, F. Meta-modelling based assembly techniques for situational method engineering. *Information Systems 23*, 7 (1998), 489–508.

[8] CARROLL, J. M. *Making Use: Scenario-Based Design of Human-Computer Interactions*. MIT Press, 2000.

[9] CEDITI. A kaos tutorial, http://www.objectiver.com/download/documents/kaostutorial.pdf, September 2003.

[10] CHECKLAND, P., AND SCHOLES, J. *Soft Systems Methodology in Action*. John Wiley & Sons, 1991.

[11] CHUNG, L., NIXON, B. A., YU, E., AND MYLOPOULOS, J. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, Boston, 1999.

[12] DARDENNE, A., VAN LAMSWEERDE, A., AND FICKAS, S. Goal-directed requirements acquisition. *Science of Computer Programming 20* (1993), 3–50.

[13] DAVIS, A. M. *Just Enough Requirements Management: Where Software Development Meets Marketing*. Dorset House, 2005.

[14] ERIKSSON, H.-E., AND PENKER, M. *Business Modeling with UML*. John Wiley & Sons, New York, 2000.

[15] HAGLIND, M., JOHANSSON, L., AND RANTZER, M. Experiences integrating requirements engineering and business analysis. In *Proc. of the IEEE Third International Conference on Requirements Engineering* (1998), IEEE, pp. 108–117.

[16] IEEESTD.830-1998. *Recommended Practice for Software Requirements Specifications*. IEEE, 1998.

[17] ISO/IEC. 9126: Software engineering–product quality–, 2001.

[18] JACOBSON, I., CHRISTERSON, M., JONSSON, P., AND OVERGAARD, G. *Object-Oriented Software Engineering*. Addison-Wesley, 1992.

[19] LAMI, G. Teaching requirements engineering in the small: an under-graduate course experience. In *1st International Workshop on Requirements Engineering Education and Training (REET'05)* (2005).

[20] LIN, L., NUSEIBEH, B., INCE, D., AND JACKSON, M. Using abuse frames to bound the scope of security problems. In *Proc. of the 12th International Requirements Engineering Conference (RE'04)* (2004), IEEE, pp. 354–355.

[21] NAKATANI, T. Improving the engineering mind in eliciting requirements. In *Proc. of the 2nd International Workshop on Requirements Engineering Education and Training (REET'07)* (2007), pp. 37–41.

[22] NAKATANI, T., AND FUJINO, T. Role and owner based business domain analysis. In *Proc. of the Asia Pacific Software Engineering Conference'2005 (APSEC2005)* (2005), IEEE, pp. 130–137.

[23] NAKATANI, T., AND FUJINO, T. Rodan: A requirements elicitation method for a business domain by focusing on roles. *IPSJ Journal 48*, 8 (2007), 2534–2550. (in Japanese).

[24] NIKULA, U. Experiences of embedding training in a basic requirements engineering method. In *Proc. of the 17th Conference on Software Engineering Eduvation & Training (CSEET'04)* (Los Alamitos, CA, USA, 2004), IEEE, pp. 104–109.

[25] SMITH, R., AND GOTEL, O. Using a game to introduce lightweight requirements engineering. In *Proc. of the Internationale Conference on Requirements Engineering (RE'07)* (Los Alamitos, CA, USA, 2007), IEEE, pp. 379–380.

[26] YU, E. S. K. Towards modelling and reasoning support for early-phase requirements engineering. In *Proc. of the 3rd International Symposium On Requirements Engineering (RE'97)* (1997), IEEE, pp. 226–235.

[27] ZOWGHI, D., AND PARYANI, S. Teaching requirements engineering through role playing: Lessons learnt. In *Proc. of the 11rd International Requirements Engineering Conference (RE'03)* (2003), IEEE, pp. 233–241.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-505

User Stereotypes for Student Modelling in Collaborative Learning: Adaptive Advice to Trainers

Kalliopi TOURTOGLOU [a] and Maria VIRVOU [b]
Department of Informatics, University of Piraeus,
80 Karaoli & Dimitriou St. Piraeus 18534, Greece
[a] ktourtog@yahoo.gr, [b] mvirvou@unipi.gr

Abstract. Software engineering usually involves the collaboration of team members in a team. One important issue of project leaders or software engineering trainers is to form teams that will collaborate efficiently. In view of this, in this paper we describe how we have adapted the stereotype-based theory for building individual user models in a collaborative learning environment in order to provide intelligent help to trainers of UML. In this work, we describe the user characteristics included in the user models and the type of advice offered to the human trainer of the system. The user characteristics are related not only to the knowledge state of the users, but also to their personality and performance types. The advice concerns the most effective organization of the users into groups taking into consideration their level of expertise, their personality characteristics and their performance types. Furthermore, we present an empirical study among experienced trainers on the combinations of stereotypes that are considered to be the most appropriate matching of the users.

Keywords. Stereotypes, student modelling, user modelling, adaptivity, collaboration, cooperation, learning, UML, advice, adaptive help, adaptive advice, effective groups

1. Introduction

It has long been recognized that in order to build a good system in which a person and a machine cooperate to perform a task it is important to take into account some significant characteristics of people [1]. A technique for achieving this is by building individual user models that describe the characteristics of the people using the system. A widely used technique for building user models is the use of stereotypes [2]. This technique imitates the method that people in real life use in order to make inferences about others. Stereotype-based reasoning takes an initial impression of the student and uses this to build a detailed student model based on default assumptions [3].

In this paper, we describe our system AUTO-COLLEAGUE (AUTOmated COLLaborativE leArninG Uml Environment) that is an adaptive Computer-Supported Collaborative Learning (CSCL) environment for UML. CSCL systems constitute a category of learning environments that provide the users with communication tools in order to collaborate with each other. CSCL systems are implemented with the aim to

apply collaborative learning skills. The concept of collaborative learning, the grouping and pairing of students for the purpose of achieving an academic goal, has been widely researched and advocated throughout the professional literature. The term "collaborative learning" refers to an instruction method in which students at various performance levels work together in small groups toward a common goal [4].

In this paper, we focus on research work about how to provide help to trainers concerning the management of teams of trainees. For this purpose, we have conducted an empirical study among experienced trainers concerning the ways they manage their teams of trainees based on the trainees' respective performance types and personality characteristics.

AUTO-COLLEAGUE is an adaptive collaborative learning environment for teaching UML. It is suitable for use in software houses for training software engineers and in educational institutes for teaching students. AUTO-COLLEAGUE is a collaborative environment, which provides adaptive help and advice to the users. The adaptivity of the system is implemented through building individual user models via the stereotype-based theory. There are two kinds of users in AUTO-COLLEAGUE: the trainees and the trainer. The trainees are the users whose objective is to learn how to construct UML diagrams. The trainees can collaborate with each other through a chat system. There is a workspace to draw UML diagrams in order to experiment with their UML knowledge. Furthermore, the trainees have to run certain problems and submit a solution UML diagram. The trainer is the user that moderates and supervises the learning process. The tasks of the trainer are the definition of certain parameters of the program and the effectuation of the adaptive advice given by the system. This advice is related to the most effective collaboration schemes between the trainees and is given in the form of an integrated suggestion on the construction of groups. In order for the system to find the most effective combinations of trainees taking into consideration a variety of their characteristics uses the Simulated Annealing algorithm [5], [6]. The Simulated Annealing is an algorithm for obtaining better heuristic solutions to combinatorial optimization problems [6].

The main body of this paper is organized as follows: In section 2, first we describe related work within the area of CSCL systems and, then, we describe related work for user modelling based on user stereotypes. In section 3 we describe an empirical study that we conducted among trainers of software engineering. The results of this empirical study led to the construction of user stereotypes. In section 4 we explain the stereotypes used by the system and the user interface of defining them. In section 5 the ways of tracing and evaluating the user errors are described. In section 6 we describe how the adaptive advice to trainers is generated and the impact of the user stereotypes in this process.

2. Related Work

2.1. CSCL Systems

There have already been developed CSCL systems, such as LECS [7], COLLECT-UML [8], DEGREE [9], CoLeMo [10], Fle3 [11], a learning environment [12] and Co-Lab [13]. In fact, COLLECT-UML [8] and CoLeMo [10] are learning environments for UML. All of these systems focus mainly on the effectiveness of their collaborative-communicative tools, such as chat systems, common workspace etc. Another common

characteristic is the content of advice they provide to the users. In particular, the advice is restricted to the knowledge state of the users (inferred by their errors), the theory topics they should study and the quality of their participation in the collaboration tools (e.g. a chat room). Additionally, some of them include mechanisms of finding the most appropriate to the individual peer helper according mainly to their level of expertise. Furthermore, the existence in the system of a human trainer is rarely found. In particular, a human trainer or moderator is included in COLLECT-UML and CoLeMo, but his/her contribution to the system is rather limited.

On the other hand, the main aim of AUTO-COLLEAGUE is not to present innovative methods and interfaces of communication/collaboration tools. It does include a message board with special functionality, but concentrates on generating adaptive advice on the most effective organization of the trainees into groups. For this reason, AUTO-COLLEAGUE builds individual user models taking into consideration, not only the level of expertise of the trainees, but their personality and performance types as well. As far as we consider, there is not yet such software of any category to produce this kind of advice and suggestions. Another characteristic that differentiates AUTO-COLLEAGUE from other CSCL systems is the contribution of the human trainer. The trainer builds the group role structure and defines the restrictions on stereotypes according to the needs of the groups.

2.2. User Modelling Based on User Stereotypes

The theory of stereotypes is a technique of building individual user models. This is the theory we have implemented in order to build the learner models in AUTO-COLLEAGUE. A stereotype represents a collection of traits [1]. Traits describe the features of the users. Each of these traits consists of a facet and a value. The facets are the attributes that the system traces and evaluates, assigning them the correspondent value. Rich [1] introduced the implementation of user modelling via stereotypes with GRUNDY, a system that suggests novels that may interest people. Since then, many other systems based on stereotyped user modelling were implemented, such as PPG [14], LIFESTYLE FINDER [15] and WEAR [16], proving the effective usability of this theory in software that is based on user models. Especially, WEAR [16] is tutoring/learning environments (student modelling) like AUTO-COLLEAGUE.

In section 4, we will explain in detail the way we have adapted the theory of stereotypes in the user modeller component of AUTO-COLLEAGUE.

3. Empirical Study on Defining the Appropriate Stereotypes

As mentioned above, AUTO-COLLEAGUE offers advice to the trainer of the system concerning the most effective organization of the trainees into groups. For this, the system takes into consideration not only the level of expertise of the trainees, but their performance types and personality characteristics as well. This is achieved through the combinations of the trainees' stereotypes.

We have conducted an empirical study on finding the most effective combinations between the user stereotypes. The empirical study included 50 experienced trainers. They were given a questionnaire in which they had to answer to questions concerning the desired combinations and the constraints on combinations between the user stereotypes of their trainees. The desired combinations are the pairs of user stereotypes

they would consider as fruitful. The constraints on combinations are the combinations they would consider as not profitable. The given questions were related to the ways they organize the trainees in their classes according to the individual characteristics they have traced. The results are illustrated in Figures 1 and 2. The trainers were also asked to justify their answers.

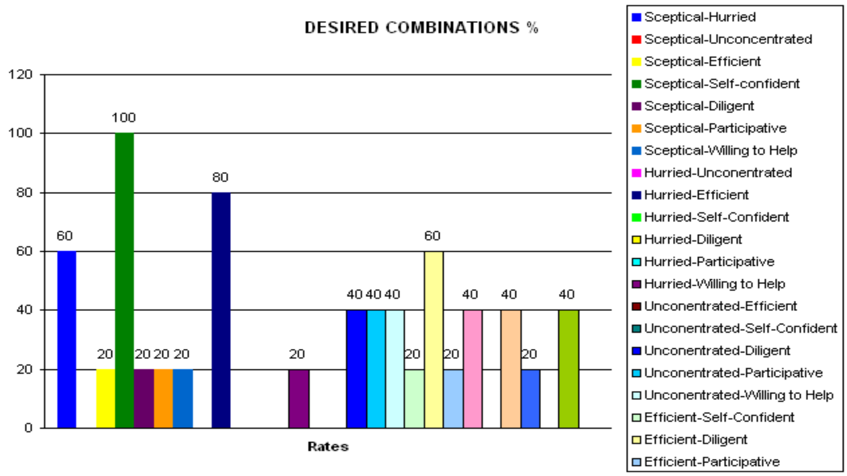


Figure 1. Results on Desired Combinations.

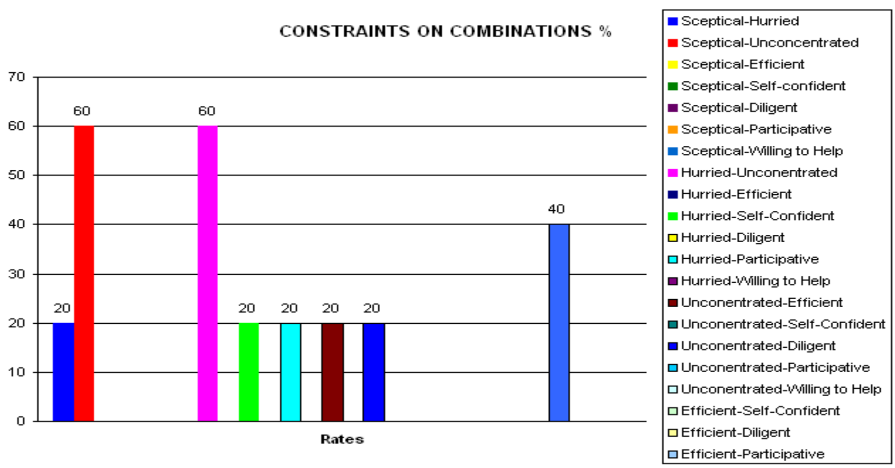


Figure 2. Results on Constraints on Combinations.

The combinations with value more than 50% were used in AUTO-COLLEAGUE through the Groups Building form.

4. The Construction of Stereotypes

The stereotypes included in AUTO-COLLEAGUE are organized into three categories: **Level of Expertise**, **Performance Types** and **Personality**. We have chosen these stereotypes as their user characteristics influence the collaboration and learning process.

The Level of Expertise stereotypes describe the amount of knowledge the user has on constructing UML diagrams. These stereotypes are: *Basics, Junior, Senior* and *Expert*. The UML knowledge is stored in the database of the system in the form of UML concepts. AUTO-COLLEAGUE includes a mechanism that tracks the user errors. Every error is associated with specific UML concept(s). In this way, the system evaluates the Level of Expertise of the user. For example, the Junior level of expertise is linked to the UML concepts: Attributes Definition and Generalization. This means that for a user to be considered by the system as junior, s/he should know these two UML concepts. The capturing of the user errors and their impact on the system will be explained later in detail.

The Performance Types stereotypes are related to the general manner the users have during the process of using a training program. These stereotypes are: *Sceptical, Hurried, Unconcentrated* and *Efficient*. Sceptical is the user that seems to need more than the average time to process the data of the problem. The sceptical user tends to make unreasonable mistakes and (relatively to his/her knowledge) the progress in his/her skills could have been faster. The hurried user usually submits the answers to problems quickly without examining their correctness and effectiveness. This results to frequent errors. The unconcentrated user seems to lose his/her abstraction during the training task and perhaps is engaged with other irrelevant tasks at the same time. This kind of performance type leads to frequent errors and increase in the average time needed to complete a task. The efficient user appears to successfully fulfill the demands of the training task. This kind of user always submits correct solutions/diagrams after a usual – or even lower than the usual – average amount of time.

Applying the definition in [17], the Personality stereotypes reflect the characteristics of the person that influence their cognitions, motivations and behaviors. These stereotypes are: *Self-confident, Diligent, Participative* and *Willing to Help*. The self-confident user believes in him/herself and his/her skills. When a person has self-confidence, s/he maintains a positive attitude even if his/her knowledge and skills are not of a high level or even if probably s/he actually is not highly esteemed by his/her colleagues. The diligent user has earnest and persistent application to the training task and makes steady efforts during the learning process. The participative user seems to like collaborating with others and have an active presence in the task elaboration. The willing to help user demonstrates good disposal to help his/her colleagues.

As explained in the stereotype theory, stereotypes are simply collections of facet-value combinations that describe groups of system users [2]. Facets are the attributes of the users, which are calculated and evaluated by the system, in order to assign the users with the appropriate stereotypes. The facets we decided to use in AUTO-COLLEAGUE are these that can provide us clues about the specific stereotypes used by the system. In particular, these facets are: *useless mouse movements and clicks frequency, average idle time, number of actions, error frequency, same error frequency, correct frequency, help utilization frequency, average time between successive help readings, advice given frequency, help given to a member/non member of the group, help request from a member/non member of the group, communication frequency* and *number of upgrades/downgrades in level of expertise*.

A system that is going to use stereotypes must also know about a set of *triggers* - those events whose occurrence signals the appropriateness of particular stereotypes [2]. A trigger is a set of rules/conditions. If these conditions are satisfied/dissatisfied for a user, then the corresponding stereotype will be activated/deactivated. For the example of the *Hurried* stereotype, we have defined the HURRIED_TRIG. The rule of this

trigger is the simultaneous satisfaction of the facet values cited in the previous section (number of actions = 5, correct frequency = 1, error frequency average idle time = 1, useless mouse moves = 3, help utilization frequency = 1 and average time between successive help readings = 1).

The rules of a trigger may concern not only the facet values, but also the activation of a stereotype. For example, a trigger in our system is the EFFICIENT_EXPERT_TRIG, which will activate the *Efficient* stereotype because the *Expert* stereotype has been activated. This represents the inference that if a user is expert, then s/he is probably efficient too.

The core of the user modeler is its model of an individual user. This model, called the *User Synopsis* or USS, is built by combining the direct information provided by the user, direct inferences from the user's actions, and predictions made by stereotypes that are deemed appropriate for this user. The information in the USS can then be used to guide the rest of the system [2]. In AUTO-COLLEAGUE, the USS is a structure that describes the stereotypes assigned to the users and the inferences that led to these assignments. The inferences can be both facet-value pairs and triggers. Not only the user models in effect are stored, but also the updates in them in the form of historical records. These updates are useful to the system for making further inferences about the user.

The stereotypes and the parameters that affect them (facets, triggers) are not hard coded. The trainer can change them through the appropriate forms of AUTO-COLLEAGUE: the form of defining the parameters of the stereotypes (Figure 3) and the form of groups building.

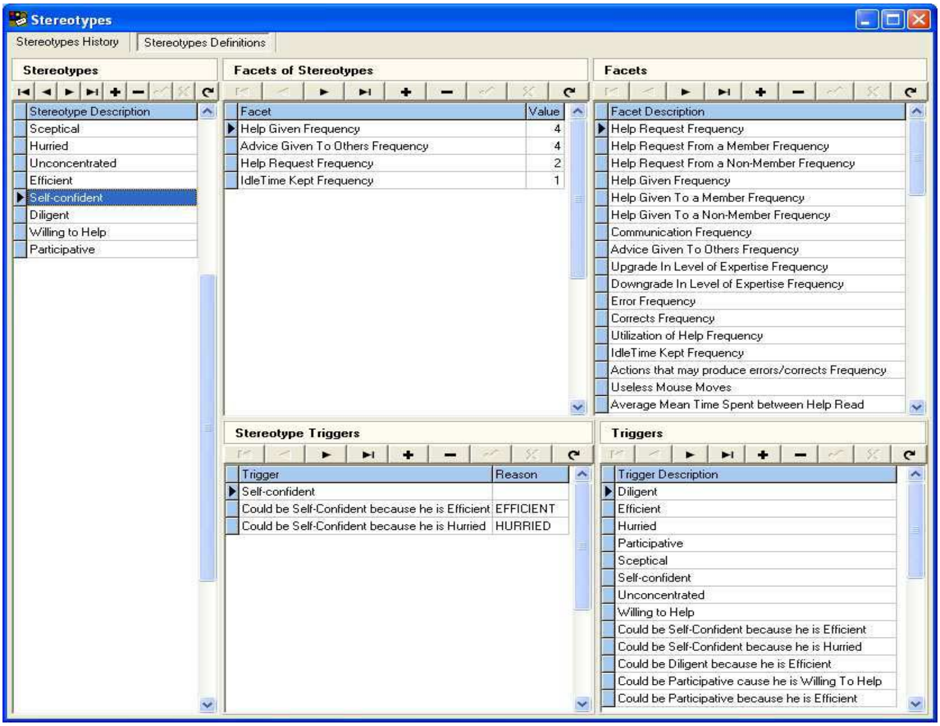


Figure 3. The form for defining the stereotypes.

On the left part of the Stereotypes' Definitions form the stereotypes that the system uses are listed. On the right part of the same form, all the facets and the triggers are listed. Whenever the trainer scrolls through the stereotypes, on the middle part of the form s/he can view (and modify) the facets and the triggers of the current stereotype.

5. Tracing and Evaluating User Errors

One of the problems we faced during the analysis of our system was the procedure of the automated capture of the user errors without the intervention of a human. The user errors are useful to the system as their frequency, type and the knowledge to which they are linked can indicate the stereotypes of the user. The method that we decided to use for tracing user errors was to provide users with certain problems/tests with a multiple-choice system of answering. The form of tests is illustrated in Figure 4.

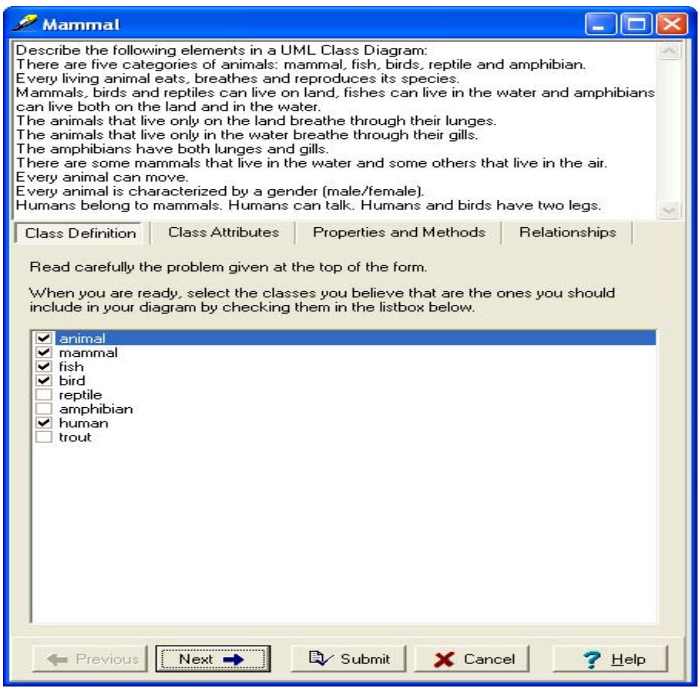


Figure 4. The "Run Test" form.

On the upper part of the form, the trainees can view the problem description. Then, the first step will be to check the classes from a given list on the lowest part of the form. This list contains both correct and wrong classes. Afterwards, the trainees will have to define the class attributes for the classes they have just checked. The next step is the definition of the attributes and methods of the checked classes. Again, they do not complete the names of attributes and methods by typing. They are given a list of attributes and methods. The last step is the definition of the relationships (the generalizations and associations) of the class diagram. After they have finished

describing the UML class diagram and pressing the “Submit” button, the proposed solution is checked and evaluated by the system. The errors of the trainee are shown in the form of text and the correct diagram is drawn.

All the errors along with supplementary information (such as the exact time and session) are stored in the database. The errors contain critical information about the users in many aspects. First of all, the facets *error frequency, same error frequency* and *correct frequency* determine the stereotypes assigned to the user. Furthermore, the level of expertise of the users is almost exclusively evaluated through their wrong and correct answers. Another utilization of the user errors is the extraction of the help topics the user will be advised by the system to study.

The domain knowledge of the system, that is UML, and the types of errors are stored in the database. Every error is linked to specific UML knowledge. This means that making a specific error indicates the lack of knowledge in particular UML topics.

6. Generating Adaptive Advice to Trainers

As already mentioned, AUTO-COLLEAGUE generates adaptive advice to the trainer concerning the most effective organization of users into groups. To achieve this, the system traces the users’ stereotypes and concludes to the most appropriate schemes of users according to the trainer’s definitions of **the groups’ structure** and **the constraints and desires on combinations of stereotypes**.

The groups’ structure includes the definition of the groups and the pattern of roles. In Figure 5, the form of defining these parameters is illustrated. In this example, the trainer has defined three groups. In Team 1, s/he needs two junior students, one senior and one expert student. In Team 2, the structure is one junior and two senior students. In Team 3, the structure is two juniors, one senior and one expert student. This is only an example of groups’ structure definitions. The trainer can change this depending on the amount and quality of the trainees.

The constraints and desires on combinations of stereotypes are those explained in sections 3 and 5.

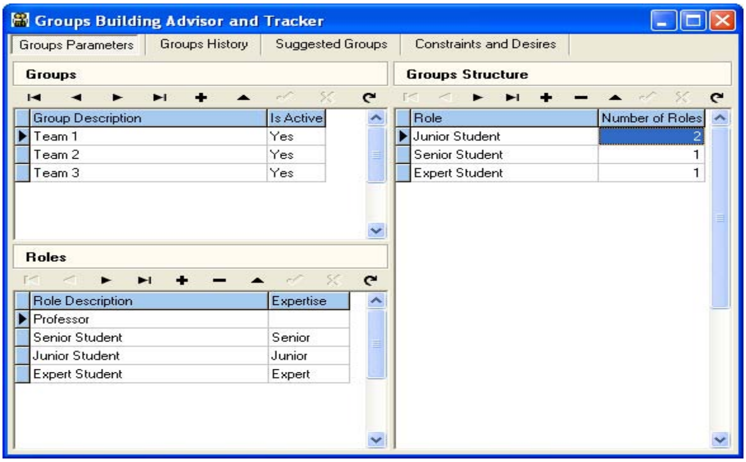


Figure 5. The Form of Defining the Groups Parameters.

There is a special form in which the trainer is shown the groups that are suggested by the system. By pressing the appropriate button, the system evaluates the above-mentioned parameters of stereotypes and groups' structure and searches for the best combinations of users into groups. This is done using the Simulated Annealing algorithm [6]. The best result is shown as a tree view, where the hierarchically first nodes represent the groups. On the right part of the same form an evaluation report is shown. This evaluation report indicates the success of the suggested groups of users. It is consisted of the number of the failed groups, failed combinations, successful groups and successful combinations. The failed/successful groups are the number of groups' structures the system failed/achieved to create. The failed/successful combinations are the number of the desired combinations between stereotypes or constraints on combinations of stereotypes the system failed/succeeded to include. It is possible that in the suggested groups there will not be any successful group/combination or even there will be failed groups/combinations. This is because the number of trainees and their roles may not produce in any possible way the groups in the structure that the trainer has defined.

7. Conclusions

One important issue in the use of CSCL systems is the achievement of an effective collaboration between the trainees. A method to fulfill this can be by organizing the groups of trainees in the most effective way taking into consideration not only their level of expertise, but also other characteristics such as their personality and their performance types. This is our aim in AUTO-COLLEAGUE, a CSCL system for teaching UML, which provides adaptive advice to the trainer of the system. The advice concerns the most effective organization of trainees into groups. The generation of this advice is a procedure that evaluates the trainees' characteristics through its user modeller component. The user modeller is implemented according the stereotype theory. The stereotypes used in our system include the level of expertise, the personality characteristics and the performance types of the users. The best solution proposed to the trainer is obtained using the Simulated Annealing, a metaheuristic algorithm used in combinatorial problems.

8. References

[1] Rich, E: "Users are individuals: Individualizing user models", Journal of Man-machine Studies 18(3), pp. 199-214 (1983).

[2] Rich, E.: "User Modeling via Stereotypes", International Journal of Cognitive Science, Vol 3, pp. 329-354 (1979).

[3] Kay, J.: "Stereotypes, Student Models and Scrutability", Lecture Notes In Computer Science, Vol. 1839, Proceedings of the 5th International Conference on Intelligent Tutoring Systems, pp. 19-30 (2000).

[4] Gokhale A.A.: "Collaborative Learning Enhances Critical Thinking", Journal of Technology Education, Vol. 7(1) (1995).

[5] Metropolis, N.A., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E.: "Equation of state calculations by fast computing machines", Journal of Chemical Physics 21, pp. 1087–1092 (1953).

[6] Kirkpatrick, S., Gelatt, C.D. and Vechhi, M.P.: “Optimization by simulated annealing”, Science Journal 220, pp. 671–680 (1983).

[7] Rosatelli, Marta C. and Self, J.: “A Collaborative Case Study System For Distance Learning”, International Journal of Artificial Intelligence in Education, Vol. 14, IOS Press,pp. 1-29 (2004).

[8] Baghaei N. and Mitrovic A.: “Collect-UML: Supporting Individual and Collaborative Learning of UML Class Diagrams in a Constrain-Based Intelligent Tutoring System”, Lecture Notes in Computer Science, Vol. 3684, Springer Berlin/Heidelberg, pp. 458-464 (2005).

[9] Barros B. and Felisa Verdejo M.: “Analysing student interaction processes in order to improve collaboration. The DEGREE approach”, International Journal of Artificial Intelligence in Education, Vol. 11 (2000).

[10] Chen, Weiqin, Pedersen, Roger Heggernes and Pettersen, Øystein. CoLeMo: “A collaborative learning environment for UML modelling”, Interactive Learning Environments, 14:3, pp. 233-249 (2006).

[11] Muukkonen, H., Hakkarainen, K and Lakkala, M.: “Collaborative technology for facilitating progressive inquiry: Future learning environment tools”, In Hoadley, C., Roschelle, J. (eds.): Proceedings of CSCL'99, Stanford University, pp. 406-415 (1999).

[12] Or-Bach, R. and Van Joolingen, W.R.: “Designing Adaptive Interventions for Online Collaborative Modeling”, Education and Information Technologies 9:4, Kluwer Academic Publishers, pp. 355-375 (2004).

[13] Martínez Carreras, M. A., Gómez-Skarmeta, A.F., Martínez Graciá, E., and Mora Gónzalez, M.: “Colab: A Platform Design for Collaborative Learning in Virtual Laboratories”, Technology Enhanced Learning, pp. 95-109 (2005).

[14] Ardissono, L, Gena C., Torasso, P., Bellifemine F., Difino A. and Negro B.: “User modeling and recommendation techniques for electronic program guides. Personalized Digital Television. Targeting programs to individual users”, L. Ardissono, A. Kobsa and M. Maybury editors, Kluwer Academic Publishers (2004).

[15] Krulwich, B.: “LIFESTYLE FINDER intelligent user profiling using largescale demographic data”, AI Magazine, 18(2) pp. 37–45 (1997).

[16] Virvou, M. and Moundridou, M.: “Student and Instructor Models: Two Kinds of User Model and their Interaction in an ITS Authoring Tool”, In: Bauer, M., Gmytrasiewicz, P. & Vassileva, J. (eds.): User Modeling 2001: Proceedings of the 8th International Conference UM2001, Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), Vol. 2109, Springer, Berlin, pp. 158-167 (2001).

[17] Ryckman, R.: “Theories of Personality”, Belmont, CA: Thomson/Wadsworth (2004).

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-515

Project Management in Information System Development Education

Toyohiko HIROTA
Faculty of Information Science, Kyushu Sangyo University,
2-3-1 Matsukadai, Higashi-ku, Fukuoka, 813-8503 Japan

Abstract. We have taught a laboratory course on system development at the School of Information Science of Kyushu Sangyo University since 2004. Based on our experience, we decided to address various problems of the education in information system development. We categorized these problems from three viewpoints. First, from the viewpoint of system development, the specific project and the method of development are important. Second, from the viewpoint of the project laboratory, it is difficult for students to do their projects effectively. Third, from the viewpoint of project management, it is a problem for both teachers and students to follow the progress of a project and to maintain control of it. In 2007, in order to solve these problems, we made several improvements, including adding Personal Software Process and Earned Value Management.

Keywords. Information System Development, Information Technology Education, Project laboratory, Personal Software Process, Earned Value Management

Introduction

We have taught a laboratory course on system development at the School of Information Science of Kyushu Sangyo University since 2004. The summary of the course and our previous experiences with it are described in this paper.

The system development laboratory is executed as a required subject, called "Social information system laboratory," at the second term of the third year. As a rule, because one teacher is assigned to one student enrolled in the laboratory, ten students or less actually attend the system development laboratory at one time.

The student is organized into groups of 3 to 5 students, which usually results in two groups. Each group receives a development task for a Web system using Java from a virtual customer, and accomplishes the task as a system development project.

The target project has been a small-scale bookstore, which handles books, publishers, members, and orders, etc. Figure 1 shows the software architecture of the system. It conforms to the model view controller (MVC), and the screen output is the view layer described with JSP (JavaServer Pages). The control layer, where the input from the user is handled, is coded as Java Servlets. In the model layer, where data is managed, each data item is processed as a Bean (JavaBeans) so that JSP may easily access the attribute.

The first laboratory was executed at the second term of 2004. At this time, the number of students was 19 because it was a two-laboratory combination, and four groups of 4 to 5 students were organized. One group transferred to another project topic for a spe-

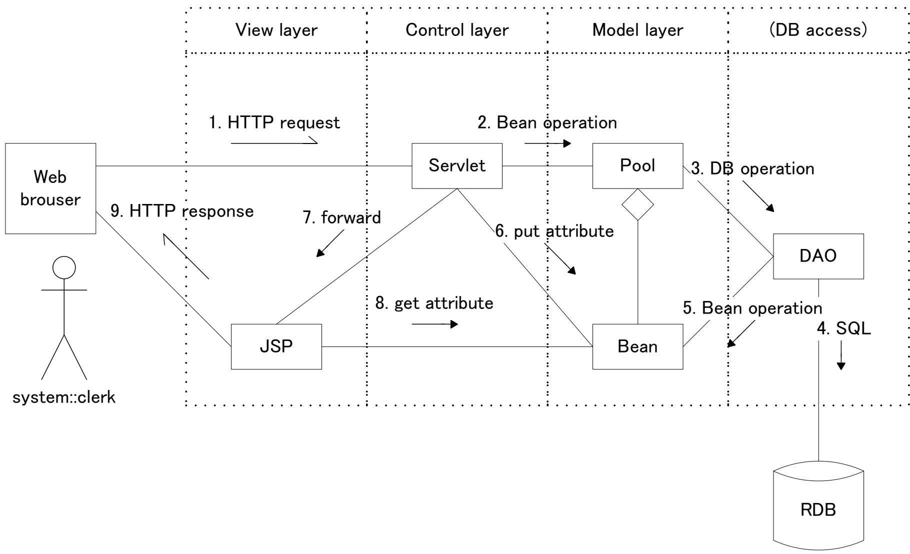


Figure 1. Software architecture of the bookstore system

cific reason, and one of the other three groups were not able to start implementation of the system. Because it was feared that time was insufficient, given the original teaching material[1], the content was reduced to less than half. Nevertheless, the students could not complete the system. There were two causes of the failure. First, because too much time was required to install the teaching material, the teacher could not confirm the program beforehand. Second, because there was no assistant, the teacher could not guide each group individually.

The system development laboratory was not executed at the second term of 2005. Therefore, the plan in the next year was reconsidered. In order to complete the assigned topic, the idea was that the content of the laboratory should be reduced, referring to the experience of the students who transferred to another topic. Part of the original system was developed on the teacher side, which reduced the part that the students would have to develop in the laboratory.

The laboratory was executed at the second term of 2006 by eight students organized into two groups, according to the plan prepared in the previous year. Moreover, one student of the fourth grade offered detailed guidance to the students as an assistant. In one group, the student who had been playing a leadership role dropped out, so all the intermediate results of the group were lost and the group fell into disorder; however, they finally completed the implementation on time. However, both groups developed a flawed system which could only handle 30 data or less. From the beginning, the assistant himself did not understand the system, and the teacher did not invest enough time to it. Judging from the fact that even the team where the intermediate result was lost could complete the system, it seems that the content of the laboratory was not overwhelming.

[1]The material was provided by NS Solutions Corporation.

Table 1. Outline of the target program in 2004

Layer	Language	Files	LOC[2]
View layer	JSP	17	476
Control layer	Java	10	438
Model layer	Java	3	289
Database layer	Java	4	379
Others	Java	1	94
Total		35	1676

Table 2. Outline of the target program in 2006

Layer	Language	Files	LOC
View layer (for Member)	JSP	17	476
View layer (for Order)	JSP	24	626
Total		35	1102

1. Problems in information system development education

The problems that emerged from the previous execution of the system development laboratory are categorized here from three viewpoints: system development, the project laboratory, and project management.

1.1. Problems in system development

The outlines of the target program in 2004 and in 2006 are shown in Tables 1 and 2. The number of files for both is 35, but the control layer, the model layer, and the data base layer, etc., are used in 2004, whereas the programs in 2006 are all in the view layer. LOC of the 2006 project was about two-thirds of LOC of the 2004.

As described in the previous section, the students could not implement the system at all in 2004. Nevertheless, the students could implement and test the system to some extent in 2006, although it contained a mistake. It does not seem that the difference of the program scale alone was a factor. If the only reason was that the scale was too large, the previous year's students would be able to start the implementation and testing of the system. However, they could not start to implement the system. Actually, the students did not seem to understand the software architecture for two reasons: 1) there was not enough time to explain the software architecture of the system, and 2) the students were neither carefully guided nor were their questions answered.

The students completed the system in 2006, but the system of each group was also flawed. This problem occurred because the students did not sufficiently understand the use of a function of an existing part of the system for realizing user requirements. Based on the experience in 2004, we limited the development task to the view layer, which made the student mistakenly assume that they could easily develop a system. It seems that the students finished their project only by copying the view layer of the member function of the existent system. Another problem is that the teacher and the assistant could not sufficiently check the requirements of documents such as the screen transition diagrams. It seems indispensable that we should guide the students regarding the requirements specifications for their implementation.

1.2. Problems in the project laboratory

There are various problems for students carrying out a project in the system development laboratory, especially when they must complete a project as a group. It is not easy to make

[2] It is lines of code: the number of lines except comment lines and blank lines.

the students recognize the importance of the quality, cost, and delivery date of the project. For example, it is also difficult to make students maintain the quality of the system, even though it is possible to meet the delivery date by setting the report deadlines. A report could be submitted even when it did not satisfy the required quality. The timeframe of related lectures is provided on the course schedule, and class periods are provided for working on the project. However, there was a limitation for the time when the teacher could report the grades of the class.

The cost is how much time the student spent at the system development laboratory. This has been almost restricted by the timetable. If the student earnestly worked on the laboratory, the cost would be reflected; otherwise, it did not have as much meaning.

For carrying out a project appropriately, it is necessary to plan the project far enough in advance, but the planning was usually insufficient. It is typical that a plan is not followed and the students' roles do not follow their assigned function-each works arbitrarily, an excellent student works hard alone, all the members do the same thing together, and so on. Even if such a situation takes place, the system development laboratory can be finished, even though the students do not experience the essence of the project laboratory.

A difficult problem emerged if a student member of a project dropped out. It is difficult to add a new member because the project is in a given class. Therefore, the content of the laboratory may be reduced for the remaining members to complete the project in the limited time. However, even the intermediate product might be lost in the worst-case scenario of a member drops out. We should guide the students so that this does not happen.

1.3. Problems in project management

It is aother issue for the students to study and experience project management, even if they carried out a project. From the viewpoint of project management, the system development laboratory projects both in 2004 and 1006 were not managed appropriately. There are two causes for this situation. One is that it is difficult to schedule a project appropriately because the student's ability and behavior cannot be understood enough beforehand. This does not always mean that the teacher only cannot sufficiently understand the situation. Even if the students make an estimation by themselves, it becomes an improper estimate.

Another problem is monitoring the progress of the project. Even if the schedule is planned for every day and it is checked, we know only that it is on schedule or it is delayed. It remains vague how long the project delays will last, why delays occur, and how to do to recover from delays. It seems necessary to introduce the technique of appropriate project management. However, time is limited, and it is difficult to spend a lot of time explaining project management in addition to system development.

2. System development laboratory

The problem described in Section 1 emerged from the experience of the system development laboratory in 2004 and 2006. Then, the form of the laboratory was completely reviewed with the goal of solving these problems in 2007.

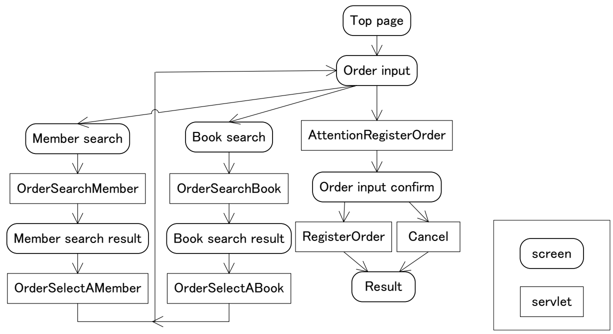


Figure 2. Screen transitions of order input

2.1. Target program

In 2007, we required the students to develop the control layer of the order function. This is because 1) the view layer is nearly described by HTML, which is rather different from usual program development, and 2) the system development becomes totally incorrect if the students design wrong screen transitions in the view layer.

The screen transitions are an important influence on the function and ease of system use. It is necessary not only to understand the demand of the user to design an appropriate screen transition but also to understand the function of an existing part of the system. However, it is not easy to make the students understand this in the situation in which the laboratory time is limited. Therefore, we decided to give the screen transitions as a predetermined solution. In actuality, we adopted the following process. 1) The students could see and operate the actual completed system. 2) All the necessary JSP programs in the view layer were deployed in advance[3]. 3) The screen transition diagrams were given.

The specification of the control layer is completely defined from the given view layer. To demonstrate this, the corresponding servlets of the control layer are described in the screen transition diagram (The order input part is shown in Figure 2). Moreover, a table that shows the correspondence with existing program files is given to the students. These files define indirectly the internal specifications of the programs to be developed. As a result, the student is guided to developing the program with the specified functions.

2.2. Development document

In the system development laboratory, the students are required to prepare documents concerning requirements specifications and designs so that they experience the entire system development. In 2004, the students were required to prepare numerous documents such as a domain model, sequence diagram, and collaboration diagram. In 2006, the number of documents was decreased to ensure the completion of implementation, and only three documents were required (data flow diagram, screen transition diagram, and ER diagram). In 2007, we reexamined these and decided to discontinue the data flow diagram and add the requirements specifications. These documents are explained as follows.

[3]Because it is necessary to put the JSP programs in a prescribed place on the Tomcat Web server, we put them in a development directory for the students beforehand.

1. Requirements specifications: they are mainly use case description and include the following items.
   (a) Outline
   (b) Precondition
   (c) Postcondition
   (d) Actor list
   (e) Event sequence
   (f) Input and output
   (g) Business rule
   (h) Occurrence frequency and pattern

   Though the students had studied neither use case use nor use case description beforehand, they could identify them with comparative ease by referring to existing specifications, such as Book and Publisher. However, there were several problems, such as a peculiar sequence of Order not appeared in Book and Publisher, and unmodified copies of unknown descriptions remained in the students' documents.
2. Screen transition diagram: The screen transitions related to order processing were described. Therefore, the students were not required to design the screen transitions, but they had to confirm the screen transitions by activating the actual system, and they had to put the transitions into documents.
3. ER (Entity Relationship) diagram: The existing ER diagram of Member, Book and Publisher, and the database schemas of Member, Book, Publisher and Order are given, and the students were required to make a modified ER diagram, in which Order was added. The students were not required to make up the database design (conceptual design), but to convert the database schemas to an ER diagram such as reverse engineering. Order in an ER diagram may be an entity or a relationship. In this laboratory, we directed the students to deal Order as an entity.

2.3. *Execution of the project*

As each student's role had no meaning in the project in past laboratories, we decided to assign a task to each student in 2007. The task dependency diagram shown in Figure 3 was given to the students because the laboratory time was limited. However, the students could decide the sequence of the tasks by themselves. In Figure 3, the name of the task (name of the module to be developed), the content of the task (function of the module to be developed), and the expected working hours are described. Moreover, the task is classified into the phases[4] of planning, design, code and compilation, testing, and postmortem according to the process script of PSP (explained in Section 2.5).

2.4. *Project management*

In general, the delivery date, quality, and cost are important in project management. In this laboratory, detailed specifications of the modules were given. Moreover, the programs that are the basis of the modules to be developed were also given. Therefore, qual-

[4] Actually, the code and compilation are separated in PSP.

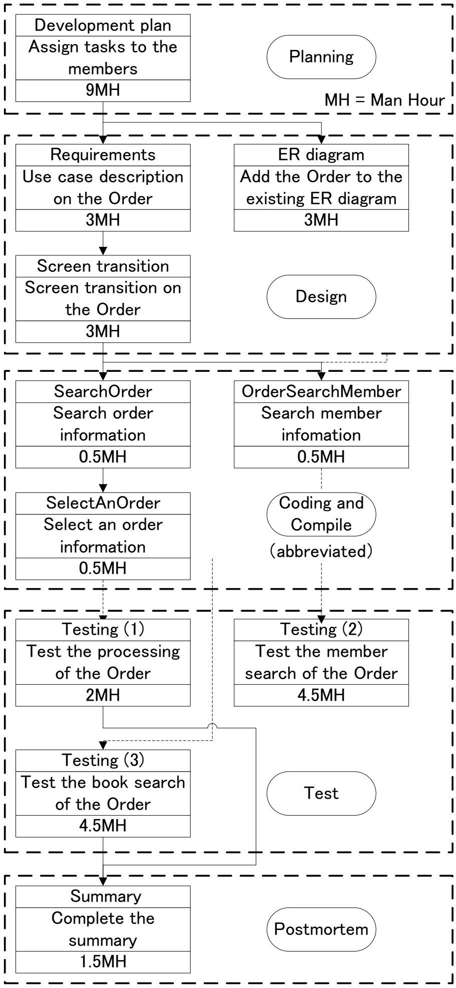


Figure 3. Task dependency diagram

ity control seemed to be accounted for in this laboratory. There was no possibility of changing the cost because it depends only on working hours, which are limited by the laboratory time in the student's timetable. All things considered, the delivery date and a status of each task influencing them became important in this laboratory.

First of all, to understand and record accurately the progress of the laboratory, Personal Software Process (PSP) was introduced. PSP is a technology that helps us to understand our individual software development abilities more accurately through the repetition of software development. Moreover, PSP presents various techniques to improve our abilities. The laboratory is offered only once to each student, which seems improper from the PSP standpoint because there no repetition of software development. However, the baseline process that is the beginning of PSP requires students to record and summarize the log, which becomes useful also for understanding the progress of the laboratory project. Usually, it is not easy to persuade students to record their logs, but in our labo-

Table 3. Project plan summary of PSP

Program Size (LOC)	Plan	Actual
Base		626
Deleted		0
Modified		350
Added		0
Added and Modified	125	350
Total Size		626
Time in Phase (min.)	**Plan**	**Actual**
Planning	540	544
Design	540	471
Code	390	389
Compile		239
Test	660	2219
Postmortem	90	164
Total	2220	3944

Defects Injected	Actual
Planning	0
Design	0
Code	12
Compile	0
Test	0
Total	12
Defects Removed	**Actual**
Planning	0
Design	0
Code	0
Compile	6
Test	6
Total	12

ratory, we judged it possible for students to achieve an accurate log record because the number of students is relatively small.

Because past accumulation data doesn't exist for PSP, the progress of the project cannot be understood from the PSP data. Therefore, we decided to introduce Earned Value Management (EVM) as another technique. EVM is a technique in which project schedules, work hours, and completed tasks are all expressed as a monetary value, and then used for project management. The status of the project can be understood in an easy line chart of two dimensions by converting everything into an amount of money. Moreover, we can easily understand the cause of the project delays, such as a shortage of the labor or underestimation of tasks.

2.5. Personal Software Process (PSP)

Personal Software Process [1] (PSP) is a technology to improve the individual ability of a person to be engaged in software development. PSP was developed by the Software Engineering Institute of Carnegie Mellon University. PSP has three stages [5] (PSP0, PSP1, and PSP2). Moreover, there are PSP0.1, PSP1.1, and PSP2.1: each of these strengthens each stage of PSP. In this laboratory, the processes were generally carried out in accordance with PSP0.1.

Table 3 shows the project plan summary of one of the two groups that participated in the system development laboratory. The Reused row and To Date and To Date % columns are omitted here because they are not applicable to this laboratory.

The program size, base, deletion, modification, and addition of programs are measured. In PSP, program size is measured by LOC (lines of code), which excludes blank lines and comment lines in the source program. In this laboratory, we used cloc [6] to measure LOC. The total number of lines of modifications and additions are presumed in the

[5] There was a PSP3 in a previous document [2], but it was replaced with Team Software Process (TSP) in recent documents [1].

[6] http://cloc.sourceforge.net/

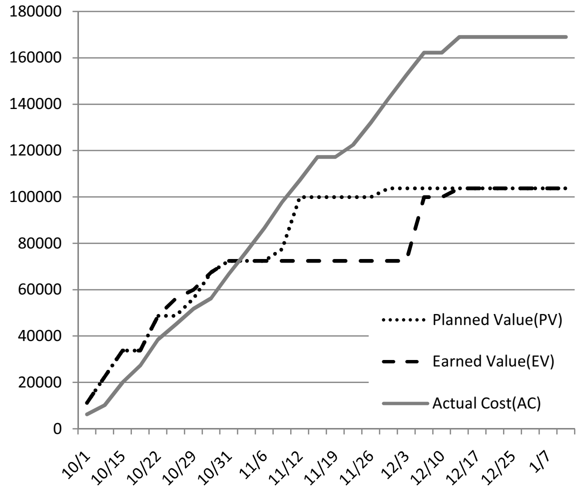


Figure 4. EVM graph

plan column of program size. We assumed that 20% of the corresponding existing program would be modified and added in this laboratory. It is necessary to calculate the deletions, modifications, and additions in the Actual column. Actually, it is not easy to assess these values because various trial and error is done during development. Therefore, these values were assumed by comparing the former code with the completed code. We used WinDiff[7] for the comparison.

In a comparison of the Actual column with the Plan column for Program Size, the total of the modifications and additions is nearly three times larger in the Actual than in the Plan column. The Plan value was measured when the senior students carried out the same project, and so it is necessary to examine in detail the cause of why such a difference occurred.

The Plan column of the Time in Phase was completed based on the man-hours described in the task dependency diagram (Figure 3). The Actual column of the Time in Phase, Defects Injected, and Defects Removed were completed based on the time recording log and the defect recording log.

In a comparison of the Actual and Plan columns of Time in Phase, there is a big difference in the Test phase. In the Plan, three tests were assumed to be performed by one person, but actually all the members together performed all the tests one by one, which seems to be the main cause of the difference. This is because all working hours were totaled in this laboratory, even though the original PSP uses only an individual record. In the future, it will be necessary to examine how to handle this case.

The total number of Code defects was 12, which is not many. Nevertheless, the students spent a long time in the Test phase. This means that the students' ability to remove defects is considerably low. In future tasks, we need to improve the students' ability.

[7] http://www.vector.co.jp/soft/win95/util/se120332.html

2.6. Earned Value Management (EVM)

In Earned Value Management (EVM), the Planned Value (PV), Earned Value (EV), and Actual Cost (AC) are measured [3].

Figure 4 shows an EVM graph of one of the two groups that participated in the system development laboratory. PV is constant at the two points because the plan included some marginal time. PV is a definite value in the last weeks because the students performed tasks not measured in EVM, such as making presentation materials or reports. EV of the latter half is a definite value for a while, though it is the same value as PV at first. This is because more time was spent than expected in the test phrase. AC is a little lower at first than PV because the actual working hours were fewer than three hours, which is the allotted time for the laboratory each week. AC increases on the way and exceeds PV. This indicates that too much time was required for testing, but it does not indicate whether the students delayed the schedule.

Measurement mistakes often occurred when the students actually applied EVM, even though EVM is an extremely simple calculation. However, it was possible to correct EVM easily if the mistake was found, because the time recording log was available due to the use of PSP.

3. Conclusion

In this paper, based on the experience of the system development laboratory, the problems of information system development education are classified into three viewpoints: system development, project execution, and project management. Considering these problems, we reduced the target program tasks, and introduced PSP and EVM in 2007.

It is easy to follow the students' progress by introducing PSP and EVM. Moreover, the EVM graph is useful for enabling the students to understand their own project situation. However, mistakes often occurred in the project plan summary of PSP and the numerical values of EVM. In the future, we intend to guide the students in measuring these values correctly.

The original purpose of PSP is to measure every single person's data continuously. However, in the system development laboratory, we think we will be able to use PSP to predict, to some degree, the progress in the coming year based on the results of the last year.

References

[1] Watts S. Humphrey. *PSP: A Self-Improvement Process for Software Engineers*. Addison-Wesly, 2005.
[2] Watts S. Humphrey. *A Discipline for Software Engineering*. Addison-Wesly, 1995.
[3] Project Management Institute. *A Guide to the Project Management Body of Knowledge*. Project Management Institute, third edition, 2004.

Special Session on Advances in Software Technologies and Cognitive Systems

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-527

Special session on Advances in software technologies and cognitive systems

Organized by

Dr. Nancy Alonistioti
Department of Informatics, University of Piraeus

Abstract

In the recent years, an interesting blending of computing, software technologies, sensors and even telecommunications concepts has facilitated the emergence of cognitive systems targeting various domains of applications.
Artificial systems that can interpret data arising from real-world events and processes (mainly in the form of data-streams from sensors of all types and from visual and/or audio sources); acquire situated knowledge of their environment; act, make or suggest decisions and communicate with people on human terms, thereby supporting them in performing complex tasks.
Cognitive radio is a paradigm for wireless communication in which either a network or a wireless node changes its transmission or reception parameters to communicate efficiently avoiding interference with licensed or unlicensed users. This alteration of parameters is based on the active monitoring of several factors in the external and internal radio environment, such as radio frequency spectrum, user behaviour and network state.
Cognition can be defined as the act or process of knowing in the broadest sense; specifically, an intellectual or artificial process by which knowledge is gained from perception. Respective concepts have impact in the evolution and emergence of cognitive communications and future internet elements.
To this end, advances in various areas are identified towards building the knowledge and mechanisms for cognitive systems operation.

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-528

Cognitive Systems in the Scope of Modern European Strategic Priorities

Ioannis P. CHOCHLIOUROS[a] and Anastasia S. SPILIOPOULOU[b]
[a]Head of Research Programs Section, Labs & New Technologies Division
Hellenic Telecommunications Organization S.A. (OTE),
99 Kifissias Avenue, 15124 Maroussi, Athens, Greece
{*ichochliouros@oteresearch.gr*}
[b]Lawyer, General Directorate for Regulatory Affairs,
Hellenic Telecommunications Organization S.A. (OTE), 99 Kifissias Avenue, 15124 Maroussi, Athens, Greece {*aspiliopoul@ote.gr*}

Abstract. Cognitive systems are a fundamental prospect of innovation and growth activities in the European Union (and worldwide) as they are able to perceive, reason and vigorously interact in modern digital operational environments by interpreting received information and acting purposefully (and autonomously) towards achieving specific goals. Their effective use will help open up new technological and business opportunities for industry and will affect an immense variety of the underlying market sectors-domains, especially by offering major innovative opportunities and facilities, aligned to the perspective of network and service convergence. Cognitive networks constitute a very significant strategic priority for the European Union policy context and actual corresponding research effort has been identified as one of the European key priority research areas. In the scope of the present work we discuss recent European initiatives and we assess the potential motivation for further development. We evaluate essential features of innovative cognitive systems, as outlined by the European strategic framework for growth. In addition, we identify several targeted areas where further progress is expected and we analyze the impact on the market sector, by identifying several categories of market players (users, network operators, manufacturers, service and application providers). As cognitive systems have the potential to address (and to satisfy) an extended variety of both modern technical- and business-related characteristics, while being fully in line with the European policies for development, they hold great promise for future wide-scale applications.

Keywords. Autonomic communications, cognitive processes, cognitive networks/systems, cognitive radio, digital convergence, future and emerging technologies (FET), information and communications technologies (ICT), knowledge-based economy, network reconfiguration, system adaptability, system robustness, technological innovation.

Introduction

Scientific research, technological advance and innovation are at the heart of the global knowledge-based economy which constitutes a key factor for growth that strongly supports the competitiveness of companies and employment. Improving the competitiveness of the European electronic communications industry and so enabling

Europe to master and shape future developments in a multiplicity of fast developing sectors of Information and Communications Technologies (ICT) activities, is an essential priority of the contemporary European strategic framework moving towards a broader evolutionary progress, that is based on global convergence's effects. In recent years, ICT developments have gained pace to arrive at the threshold of massive growth in information society and media, made possible by extensive fast communications, connecting manifold devices.

One of the fundamental objectives of the latest European "*i2010 initiative*", that sets the core strategic priorities [1] for effectively applying various ICT-based policies in the European Union (EU), is to achieve a "*world class performance in research and innovation in ICT by closing the gap with Europe's leading competitors*" [2]. Thus, leading the advancement in selected modern ICT areas is critical to address Europe's key socio-economic challenges and to strengthen its industrial competitiveness [3]. In fact, the various challenges facing Europe's society, economy and environment are now surmountable; if managed well, they can be turned into new opportunities for Europe to grow and create more perspectives for growth and development [4]. In this particular context, knowledge is at the core of the European research agenda and underpins all its elements. Research and technology are, together with education and innovation, fundamental components of the "*triangle of knowledge*", which functions best when any among the accompanying framework conditions reward the knowledge that is put to work to the immediate benefit of the economy and society. Under this consideration, innovation can also be estimated as a broader business process connected by exploiting market opportunities for new facilities and products, services and business processes. Thus, to become the "*most dynamic and competitive knowledge-based economy in the world*" while maintaining the "*European Model*" which is implied by its specific initiatives, Europe further increases its research effort and better exploits its capacities in all related appropriate fields, especially by transforming scientific results into new products, processes and infrastructures or services [5].

1. Cognitive Systems in the European Context: From the Past to the Present….

In earlier EU research and development (R&D) Framework Programmes, the essential features of "*cognitive systems*" were addressed mainly by the "*Long Term Research (LTR)*" and "*Future and Emerging Technologies (FET)*" modules of the ESPRIT [6] and IST [7] programmes, *respectively*. Then, in the broader context of the 5th Framework Program (FP5), an IST Action Line "*Cognitive Vision*" was established to enhance traditional approaches on basic technologies and infrastructures through cognitive elements such as categorisation, object and action recognition, and learning, all potentially acting as "engines" of sustainable expansion. The original objective was to construct physically instantiated (and/or embodied) systems that could be able to recognize, understand and interact with their surrounding environment, in order to achieve human-like performance in activities requiring context- (situation and task) specific knowledge. As for the recent 6th Framework Program (FP6) initiative, "*Cognitive Systems*" has become a strategic objective (SO) in its own right and figures as such in the IST work programmes 2003-2004 and 2005-2006, respectively.

The challenge now emerges within the perspective of the actual 7th Framework Program (FP7)[1], where "*Cognitive Systems*" and/or "*Cognitive Networks*" constitute a very significant strategic priority [8]. The EU, *through this newly established set of activities* intends, *among others*, to deliver the next generation of ubiquitous and converged network (and service) infrastructures for communication, computing and media, and also addresses the promotion of excellence [9], especially by launching longer term public-private partnerships to support cooperation and technological development between universities, research centres, industry, market players and public authorities across the EU as well as with the rest of the world [10].
Such novel infrastructures, highly relevant to various industry's needs, are expected to allow the appearance of a large diversity of business models, able of dynamic and seamless end-to-end composition of resources across a multiplicity of devices, networks, providers and service domains. The coordination of the European R&D instruments will be further enhanced by focusing them on key bottlenecks such as interoperability, security and reliability, identity management, rights management and ease of use.

The fast speed of globalization procedures has exposed both the EU economy and technology to mounting competition from abroad. As well, the increasing complexity of modern society and economy places greater emphasis on various innovative systems which can deal autonomously with our needs and with the peculiarities of the environment(s) we live in and construct [11], especially by supporting properly selected investment actions, for the development of novel infrastructures. A very important prospect is focused on the fact that cognitive systems are considered as appropriate to offer an enabling technology for all sorts of corresponding applications, involving a "huge" interaction with the real-world environment. There is now a wide range of relevant systems whose functioning depends on expedient information processing. Endowing such systems with cognitive capabilities -as present, *in greater or lesser degree*, in living organisms- would make them more robust, versatile and flexible, more responsive and -*where necessary or appropriate*- also more autonomous (e.g., pro-active, self-sustaining, self-healing, etc.). Thus, such enhanced systems should display characteristics and behavioural attributes which make them either fitter for use in contexts where they are normally used (or operated) or fit for use even in contexts where they could otherwise not be used. Such systems could or should either dispose information about their environment (through classification, categorisation, recognition and description of events and processes) or, in addition, exert some control over processes in their environment (directly through immediate action or indirectly through supporting human decisions).Exerting control, implicates expedient knowledge about and some sort of "understanding" of the underlying "environment".

The core challenge is to develop and extend suitable systems' engineering methods to deal with open-ended and frequently altering real-world environments. A primary aim is to expand system capabilities so that to respond intelligently to "gaps" in the system's knowledge and to situations (or contexts) that have not been specified in its original design. In order to fulfil this fundamental challenge, a mixture of innovative scientific theory and technology are necessitated, based on natural and artificial

---

[1] In FP7 the technology pillars for the years 2007-20011 have been defined as follows: (i) Technologies for knowledge, content and creativity - including cognition, simulation and visualisation; (ii) advanced and open communication networks; (iii) secure and dependable software; (iv) embedded systems, and; (v) nanoelectronics. The 7th Framework Programme is tailored to better meet industry's needs.

cognition, in conjunction with new systems design and engineering principles and implementations for fitting solutions which are robust and versatile enough to deal with the real world and to behave in a user-friendly and intuitive way with people in everyday situations.

2. The Motivation for Modern Cognitive Systems

With its recognized strengths in communication equipment, devices, networks and e-Services, Europe is well placed in the worldwide competition to describe, define and further develop the *network and service infrastructures of the future*. These infrastructures are expected to create significant economic opportunities with novel classes of networked applications, whilst reducing current operational expenditures. The present Internet, mobile, fixed and broadcasting networks and the related software service infrastructure need to evolve accordingly ([12], [13]), in order to allow another wave of growth in the on-line economy and society in the coming years ([14], [15]). So the EU in the context of the recently established 7th Framework Programme intends to develop the next generation of ubiquitous (and converged) network infrastructures, able to permit the appearance of diverse business models capable of dynamic and seamless end-to-end composition of resources across a multiplicity of devices, networks, providers and service domains. Research into "*cognitive systems*" has been identified as one of the European key priority research areas, and therefore currently attracts significant attention. More specifically, the particular topic of Autonomic Communications falls under the *FET* (*Future and Emerging Technologies*) scope and focuses on paradigms shifts and technology disruptors that would reshape the way we communicate and interact [16].

A *cognitive process* is a process that improves its performance through experience gained over a period of time, without complete information about the "environment" in which it operates. Performance improvements are achieved through *reconfiguration*, meaning the capability of the process to dynamically change its structure in response to dynamic changes in the environment[2]. According to the latest international practice, *cognitive systems* are systems that perceive, understand the semantics of information conveyed through their perceptual input (i.e. by "recognizing" and "categorizing" information), learn and develop through individual or other interaction with their environment (i.e. they become able to "decide" and/or to "act"), and consequently "evolve" (by suitably reconfiguring themselves) in order to achieve human-like performance in activities requiring context specific knowledge [17]. A *cognitive network* has a cognitive process to perceive current network conditions, and then plan, decide and adapt (reconfigure itself) on those conditions. Such a network can learn from these adaptations (reconfigurations) and use them to make future decisions, while taking into account end-to-end goals.

While research on artificial cognitive systems may derive "inspiration" from the study of natural cognitive systems and/or from biological intelligence (i.e. structures

[2] The concept of "environment" is broad in scope: It can range from (parts of) natural "real worlds" to (man-made) "artificial worlds". The former may be uncharted territory (outer space, distant planets), volcano slopes, a disaster area or one or several human beings; the latter include networks and/or infrastructures of all sorts but also factories, laboratories, operating theatres, offices, many other "socio-technical" constructs. Furthermore, there is an extended range of hybrid or "mixed", environments.

and functions in biological systems), "*cognitive systems*" is now a separate scientific discipline relevant to specific systems' architectures that combine: Perception (via the provision of models leading to the selection of behaviours or to the execution of several actions); action (this implicates, for example, control, communication/interaction and change of internal system's state); reasoning (a feature which is often required for coordinating perception and action, mainly for selecting transformations, behaviours, plans or for adapting plans and/or for generating new plans); learning (by emphasizing on modes of learning (such as learning categories, competences, concepts, affordances, etc.) or by integrating new and old knowledge/skills coherently); motivation and communication. Any generalized system's "success" (e.g., in terms of achieving expected operational "goals") in complex environments depends on: (i) Precision and robustness of perception (e.g., through recognition, analysis and understanding of patterns in space, time and space-time) and (re-)action; (ii) the ability to acquire and organize relevant knowledge, to "cope with change" and to "act in time". Depending on the frequency of change this may entail more or less strong real-time requirements on a system's reaction and (possibly anticipatory) action.

As a scientific discipline, *Artificial Cognitive Systems* seek to provide an enabling technology for several sectors (for example, modern communications networks, automation and robotics, natural language understanding, man-machine interaction and complex real-world systems) [18]. However the relevant concept is not about applications in any of these set of tasks (or domains) only. Cognitive systems will require convergence of action, perception and reasoning. Actions may involve applying and controlling suitable technical devices and solutions. Alternatively, they can have the form of communicating or interacting with humans. Actions may also have the form of changes to the internal state of the system, such as a change in focus of attention, with no immediate external manifestation. Usually, the cognitive capabilities of an artificial system are supported by one or several specific subsystems (as is the case for most "higher" forms of life) which may or may not be physically integrated ("embedded") within the system itself; these subsystems, *however*, must take into account the physical and architectural characteristics of the larger system, including the particular ways the larger system is "connected" to its environment. Hence there are several "layers of control" within artificial cognitive systems and between the entire system and its environment.

Today's ICT systems, without any extra intervention, cannot "learn" from experience and reason, cannot contextualize and adapt to evolutionary processes, and cannot (inter-)act based on their own observation and learning. Many ICT applications cannot be developed further if there are no new breakthroughs in systems' intelligence and engineering [19]. Overcoming such technology roadblocks opens the doors to a wide range of opportunities in new application fields such as management (and reconfiguration) of communications networks, vision/sensing systems, service robots, health and industrial robots, multimodal and multilingual interactions and many more.

3. Essential Features of Modern Cognitive Systems

Modern cognitive systems can benefit from robust and adaptable learning techniques to recognize, identify and act upon their environments (real world or digital) without having to rely on massive pre-stored data sets or on fully pre-programmed instructions.

Such systems, as opposed to "traditional" computer-based information processing systems, should enjoy a number of general (interrelated) features and traits, irrespective of their specific environments and tasks. Preferably, they should be capable of:

- Interpreting (or "*making sense of*") whatever they are poised to sense in whatever environment they are operating (including interpretation of "environmental data" in terms of abstract roles or goals, recognition of affordances, etc.).
- Possessing some form of "understanding" (or "awareness" or "consciousness") of their own role and situation in the environment they are functioning.
- Predicting future events in their environment (including, *where pertinent*, the behaviour of other "entities" operating in the same environment).
- Having targeted learning (supervised or unsupervised, through interactions with their environment) to modify the way they operate and/or to improve their performance according to given and clearly defined criteria.
- Pursuing goals (set by humans but also "by themselves", possibly inherent in their architecture/design, e.g., to bring about changes in the environment) and "*what if…*" deliberation.
- Behaving "sensibly" and "robustly" under conditions of uncertainty (e.g., in "uncharted", unpredictable environments, but also in everyday situations), e.g., through creative exploitation of novel situations in terms of previously learned regularities, through generalization and analogical reasoning.
- Offering scalability (and adaptability) across various dimensions while still guaranteeing critical control properties (this feature is among the general desirable ones considered for the related systems architectures).

In the ICT-based sector, the challenge is to develop robust cognitive systems acquiring and using knowledge for decision making. The present focus is on adaptive systems, real-time platforms and architectures permitting the development of novel computational frameworks, integrating multiple cues for modelling and capable of identifying different situations during performance.

The future prospect of cognitive systems which can perceive, reason, and interact robustly in open-ended environments is an ambitious challenge which goes beyond today's systems engineering paradigm. Present day systems engineering relies on specifying every eventuality a system will have to cope with in the execution of its task(s) and programming the appropriate response in each case. With the abundance of ever cheaper, smaller sensors, actuators and wireless links that bring systems in contact with the real world and with other systems, this approach faces several limitations.

Artificial software-based cognitive systems, as opposed to traditional machine or computer systems can be thus characterised as these coping with *novel or indeterminate* situations, which aim to achieve *general goals* as opposed to solving specific problems, and which integrate *capabilities* normally associated with people or animals such as perception, learning, reasoning, communication, and so forth. As a consequence, they should be *more robust, more adaptive, more effective and more natural*. In fact, it should be expected that such systems need to be able to respond intelligently and largely autonomously to "instabilities" in their knowledge and to situations that have not been identified in their original planning processes (that is, they must be robust and flexible), in a way that performance would not degrade when these systems are presented with unexpected data. The challenge for a system to handle

unexpected data or events ("surprises"), either within the original system's domain knowledge or beyond it, raised much interest during recent years. Furthermore, these systems need to be more effective in improving their performance as they would be able to foresee or anticipate what might happen at some point in the future, near or far. Last but not least stands the requirement for these "structures" to be more natural when operating within their specific environmental context; that is, under this latter consideration, performance should be tolerant to the ambiguity and uncertainty that is a consequence of dealing with a fast modified surrounding environment but performance should improve with time as well.

As a result, expected progress in any of the above fundamental areas, towards systems with the relevant essential characteristics can only be achieved by developing and adopting new engineering principles and approaches, based on largely common, but as yet not fully explored, scientific grounds. Engineering progress may also depend on advancing scientific understanding of what artificial systems can and cannot do, and how and why. Thus, any suggested principles are convenient to: (i) Be able to achieve general goals in a largely unsupervised way, and persevere under adverse or uncertain conditions; (ii) adapt, *within reasonable constraints*, to changing service and performance requirements, without the need for external re-programming, re-configuring or re-adjusting, and; (iii) communicate and co-operate each other (or eventually with external entities, also including people) based on a well-grounded understanding of the objects, events and processes in their environment, and their own situation, competences [20]. This also implicates the necessity for innovative theories (systems theories, software architectures, control theories, modelling theories, etc.) to allow a sufficient building of such types of systems. Besides, any attempt for further engineering progress is very strongly dependant upon advancing scientific understanding of what both natural and artificial systems can and/or cannot perform, under which terms/conditions and for which specific reason [21].

4. Targeted Areas and Impact on the Market Sector

Current research and applied technical initiatives mainly focus on sectors where distinctive and transparent progress is realized, mainly via a clear identification of practical engineering goals, able to support diverse innovative features and so to affect our everyday life. The industry sector suggests that it is of prime importance to "identify" applications which are crying out for specific types of artificial systems-networks, able to learn, evolve and act autonomously. Converged networks and modern service models are changing the way people communicate. In fact, consumers and businesses are being offered new kinds of services and applications, while new security vulnerabilities continue to emerge and network management challenges continue to grow. As a consequence, occasions to innovate in several challenging areas (such as the management of connectivity, security and bandwidth), are thus emerging, where additional support has to be provided, especially under the FP7 context. These are fields of science where long-term economic benefits are large, where there is a need for high-risk, high cost research and where it only makes sense to share such risks, and the rewards, at EU level. The response is to emphasize upon launching essential new initiatives while reinforcing existing actions of recognised impact and added value.

Cognitive networks are often quoted as being one the key next generation communications technologies [22] and are expected to lead to a much improved communication service, while providing efficient solutions to problems currently experienced by users, network operators, manufacturers, etc. The present European concern for next-generation communication network implementations and for the corresponding infrastructures considers that, *inter-alia*, these will:

- Be pervasive, ubiquitous and highly dynamic. They are expected to provide approximately "unrestricted" capacities to users, by supporting an extensive variety of nomadic interoperable devices and services, a diversity of content formats and a multiplicity of delivery modes. They are also expected to support context awareness and the dynamic behaviour essential for applications with requirements that may vary with time and context.
- Be able to adapt, under realistic and rational limitations, to altering operational conditions without the need for external intervention on several domains (such as those relating to configuration, programming, resources management, adjustments, etc.) [23].
- Guarantee robustness, resilience, trust and security compatible with networks and software service platforms reaching a complexity and scale that are (at least) an order of magnitude greater than those of today's infrastructures.
- Support networked and managed business and service convergence across a multiplicity of environments such as the home, businesses, or nomadic situations.

Cognitive networks have the potential to address (and to satisfy) almost all among the previous envisaged characteristics, and are therefore fully in line with the European policies and hold great promise for future wide-scale applications. Solutions based on the cognitive reconfigurable network concept will include technologies and products to assure that networks, network components, networked devices and applications, can all be developed and managed (i.e. configured, optimized, healed and protected) in real time [24]. Thus, cognitive networks will represent a paradigm of distributed management functionality, possible to be realized compliant with the autonomic computing paradigm. The collective cooperation of the developed autonomic components will also give way to self-healing and scalable systems that will support the potential evolution of services and the efficient coverage of user needs [25].

Cognitive (re-configurable) networks have the immense potential to restrict (or even to "delete") severe "insufficiencies" such as inter-operator dependencies, frequent infrastructure upgrades and problems arising from split network management [26]. Moreover, such networks can maximize operators' ability to have profit from economies of scale, as implied by common hardware platforms and software architectures promoting evolution of offered access solutions. In addition, they are able to enhance time-to-market performance via the promotion/adoption of new services without the requirement for upgrading the underlying infrastructure(s), and can take full advantage of return-on-investment by maximizing utilization and exploitation of the deployed resources. Thus, they can accelerate innovation through dynamic adjustment of the corresponding network parameters, so that to conform to any important appearing needs or requirements [27].

The attractiveness of the EU both as an investment location and as an area for realizing effective actual business activities depends, *among others*, on the size and openness of its markets, its transparent regulatory environment and the quality of its infrastructures. Investment in knowledge-related and in infrastructure-related issues[3] is currently a very important matter for the EU [28], as modern electronic communications infrastructures constitute a prerequisite for reaping the benefits of the digital economy. By further widening the corresponding markets, interconnected and interoperable European networks help foster international trade and fuel internal market dynamics. Moreover, the ongoing liberalisation of European network industries fosters competition and drives efficiency gains in these sectors. Important benefits from the usage of cognitive networks can be identified on several categories of "market players", including users, network operators, manufacturers, service and application providers, listed as follows:

Concerning users, they can enjoy a multiplicity of advantages including improved connectivity, enhanced quality of service (QoS), improved personalization features, service provision and adaptability in a more integrated communications environment where various technologies can be interoperated and interconnected. It should be expected that a variety of advanced services and applications should be offered at (affordable or "appropriate") prices, as a result of the relevant investments for the installation and the deployment of the corresponding network solutions.

Network operators will possess more options of serving users as well as the possibility for further increasing their corporate income. Cognitive network solutions also provide significant facilities for optimized network installation, usage and maintenance procedures. Furthermore, they can support performance and realization of more security checks including, but not limited to, authenticity, authorization, validation, integrity check, fault diagnosing, error recovery, and so forth. In addition, the realization of scalable and reconfigurable infrastructures can, *without any doubt*, optimize usage of existing resources, of any specific nature. This option can thus facilitate an easier introduction and a more effective development of new (value-added) services, together with more attractive offers of applications and technologies. Cognitive networks can contribute towards achieving higher return on investment, mainly via the reduction of both capital expenditure (CAPEX) costs and operational expenditure (OPEX) costs. It should be expected that less hardware will be required to be used, there will be a tight "coupling" between existing forms of technologies (mainly by providing support for protocols and protocol features, and by supporting dynamic insertion, replacement and configuration of protocol components), and more advantages will be provided at network level.

Manufacturers can have significant opportunities in order to: Penetrate into new wider market domains where customized solutions can be offered; accelerate their time-to-market processes, and; reduce product development costs via an increased product flexibility and reliability. The development of "plug and play" technology will ease migration to new technical standards, protocols and applications unknown as for today. Customer support will be facilitated as well.

---

[3] The dynamism of the European economy is crucially dependent on its innovative capacity. In particular, knowledge accumulated through investment in R&D, innovation and education is a key driver of long-run growth. Policies aimed at increasing investment in knowledge and strengthening the innovation capacity of the EU economy are at the heart of the Lisbon strategy for growth and employment

Service and application providers will have equal opportunities for easy deployment of new services accompanied by enhanced features, operating on open/flexible platforms and associated execution environments. They can also penetrate in new market sectors via a more efficient service operation, with reduced time-to-market procedures.

Maintaining the stability and performance levels of complex communication systems and guaranteeing their “survival” (in the sense of their ability to continue to function under critical conditions) within their operational environment is a matter of extreme importance, where further effort can be spent for the benefit of the relevant market sectors. Equally important may be activities focused on realizing autonomic computing and autonomic communication. However, the problem space of autonomic communications relates to the effective use of multidisciplinary skills, originating from different thematic areas such as software and hardware architectures and platforms, systems design, control theory, performance and control of distributed system topologies, behaviour of complex systems, societal impact of adaptive/self-learning systems, and many more.

5. Conclusion

The future prospect of cognitive systems able to perceive, reason and vigorously interact in modern digital operational environments, constitutes a technical and business challenge of major importance that exceeds current systems’ engineering concepts. Modern cognitive systems can interpret information (and other forms of properly selected data arising from real events and/or processes), and act purposefully and autonomously towards achieving well defined goals. Cognitive systems and advanced interaction technologies will help open up new opportunities for industry not only in Europe but worldwide and will be able to affect the entire underlying market sectors, especially by offering immense opportunities for growth, together with new facilities [29].

In essence, cognitive networks may form a vital part of future communication networks, as they can deal with an enormous variety of uncertainty issues (in the form of unexpected events, diverse contexts, novel situations, etc.). The goal of autonomic networking is to tackle problems at their roots, by moving intelligence to the network, introducing joint decision-making within network (thus meeting requirements for robustness, versatility and adaptability), and minimizing *-to the extent possible-* human intervention. Reinforcing leading edge research and applied collaborative activities in these domains will help extend technologies into tomorrow’s industries and markets, in various fields of potentially high socio-economic significance (i.e. industrial production, learning, healthcare, public safety, environmental monitoring, etc.). In addition, cognitive systems can extend the capabilities of users to perform several tasks, while scientific research will also improve our understanding of the mechanisms underlying artificial and natural cognition.

Thus, among the recent European strategic objectives, several priorities have been given to the optimized control, management and flexibility of the future network infrastructure, supporting the rapid evolution towards cognitive networks being capable of: (i) Enabling seamless end-to-end network and service composition and operation across multiple operators and business domains; (ii) supporting a broad variety of

service characteristics/attributes and requirements, that will be more complex than those of the already existing infrastructures, through support of programmability and dynamic features, with re-configurability of resource allocation, of protocols and routing, self-organisation and self-management; (iii) managing, in real time, new forms of ad-hoc communications with intermittent connectivity requirements and time-varying network topology, and; (iv) enabling intelligent distribution of services across multiple access technologies with centralized or distributed control. Current European R&D activities focus on selected areas of extreme importance for the future of the corresponding communications systems and/or networks, and are expected to contribute, *significantly*, towards realizing a more enhanced and developed information society, based on intelligent infrastructures that will constitute appropriate highways for the traffic of information and knowledge.

References

[1] Commission of the European Communities (2005). *Communication to the European Parliament, the Council, the Economic and Social Committee and the Committee of the Regions, on "i2010 - A European Information Society for Growth and Development" [COM(2005) 229 final, 01.06.2005]*. Brussels, Belgium: Commission of the European Communities.

[2] Commission of the European Communities (2005). *Commission Staff Working Paper on Communication form the Commission "i2010 - A European Information Society for growth and employment - Extended Impact Assessment" [SEC(2005) 717/2, 01.06.2005].* Brussels, Belgium: Commission of the European Communities.

[3] Commission of the European Communities (2006). *Communication to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the Regions, on "Bridging the Broadband Gap" [COM(2006) 129 final, 20.03.2006]*. Brussels, Belgium: Commission of the European Communities.

[4] Chochliouros, I.P., & Spiliopoulou, A.S. (2005). Broadband Access in the European Union: An Enabler for Technical Progress, Business Renewal and Social Development. *The International Journal of Infonomics (IJI), 1*, 05-21.

[5] Commission of the European Communities (2004). *Communication from the Commission on "Science and technology, the key to Europe's future - Guidelines for future European Union's policy to support research" [COM(2004) 353 final, 16.06.2004]*. Brussels, Belgium: Commission of the European Communities.

[6] *http://cordis.europa.eu/esprit/home.html*

[7] *http://ec.europa.eu/information_society/index_en.htm.*

[8] Commission of the European Communities (2005). *Commission Staff Working Document - Simplification in the 7th Framework Programme [SEC(2005) 431, 06.04.2005] {COM (2005) 119 final}* Brussels, Belgium: Commission of the European Communities.

[9] *http://cordis.europa.eu/fp7/ict.*

[10] Commission of the European Communities (2005). *Communication from the Commission on "Building the ERA of knowledge for growth" [COM(2005) 118 final, 06.04.2005]*. Brussels, Belgium: Commission of the European Communities.

[11] Commission of the European Communities (2005). *Proposal for a Decision of the European Parliament and of the Council on Establishing a Competitiveness and Innovation Framework Programme 2007-2013 [COM(2005) 121 final, 06.04.2005]*. Brussels, Belgium: Commission of the European Communities.

[12] Mitola, J. (1999). Cognitive radio for flexible mobile multimedia communications. In IEEE Press (Ed.), *Proc. of 6th International Workshop on Mobile Multimedia Commun. (MoMuC '99), Nov. 1999*, (pp.03-10), San Diego, CA. NY: IEEE Press.

[13] Mitola, J. (2001). Cognitive radio for flexible mobile multimedia communications. *J. Mobile Networks and Applications*, *6*(5), 435-441.

[14] Haykin, S. (2005). Cognitive radio: Brain-empowered wireless communications. *IEEE J. Select. Areas Commun., 23*(2), 201-220.

[15] Akyildiz I.F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002) Wireless sensor networks: A survey. Computer Networks, *38,* 393-422. [Retrieved on April 25, 2008, from *http://www.ece.gatech.edu/research/labs/bwn/sensornets.pdf*].

[16] European Commission (2007). *ICT - Information and Communication Technologies. [A Theme for research and development under the specific programme "Cooperation" implementing the Seventh Framework Programme (2007-2013) of the European Community for research, technological development and demonstration activities]. Work Program 2007-2008.* Brussels, Belgium: Commission of the European Communities.

[17] Bickhard, M.H., & Terveen, L. (1995). Foundational Issues in Artificial Intelligence and Cognitive Science: Impasse and Solution. Elsevier Scientific, Amsterdam.

[18] Stork, H.-G. (2004). *Cognition and (Artificial) Cognitive Systems: Explanatory & Exploratory Notes*. [Retrieved on March 25, 2008, from *ftp://ftp.cordis.europa.eu/pub/ist/docs/dir_e/cognition/cognotes1_en.pdf*].

[19] Pfeifer, R., & Scheier, C. (1999). *Understanding Intelligence*. Cambridge, MA: MIT Press, (pp.5-6)

[20] Maloney, C. (2006). *ICT Work Programme 2007-2008.Challenge 2: Cognition, Interaction, Robotics*. Presentation given at the IST 2006, November 22, 2006, Helsinki, Finland. Retrieved on March 25, 2008, from *http://ec.europa.eu/information_society/istevent/2006/cf/conference-detail.cfm?id=1057*.

[21] Fette, B. (2006). *Cognitive Radio Technology*. Elsevier Inc. [ISBN: 978-0-7506-7952-7].

[22] Elliott, C., & Heile, B. (2000). Self-organizing, self-healing wireless networks. *Proceedings of IEEE International Conference on Personal Wireless Communications, December 17-20, 2000* (pp.355-362).

[23] Cerpa, A., & Estrin, D. (2001, May). ASCENT: adaptive self-configuring sensor networks topologies, UCLA Computer Science Department Technical Report UCLA/CSDTR-01-0009. [Retrieved on April 12, 2008, from *http://www.lecs.cs.ucla.edu/Publications/papers/ASCENT-Infocom-2002.ps*].

[24] Arslan, H. (2007). Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems. Springer. [ISBN: 978-1-4020-5541-6].

[25] Boscovic, D. (2005). *Cognitive Networks - Motorola Technology Position Paper.* Schaumburg, Illinois: Motorola Inc. [Retrieved on February 14, 2008, from *http://www.motorola.com/content.jsp?globalObjectId=6642-9277*].

[26] Mahonen, P. (2004). Cognitive trends in making: future of networks. In IEEE (Ed.), *Proceedings of IEEE 15th Personal Indoor and Mobile Radio Communications (PIMRC) International Symposium*, Barcelona, Spain, September 05-08, 2004 (pp.1449-1454). [ISBN: 0-7803-8523-3].

[27] Harada, H. (2005, May). A study on cognitive radio and its applications. (Technical Report, SR2005-18). *IEICE Transactions on Communications Journal, 105*(36), 117-124.

[28] Commission of the European Communities (2005). *Integrated Guidelines for Growth and Jobs (2005-2008) [COM(2005) 141, 12.04.2005].* Brussels, Belgium: Commission of the European Communities.

[29] *http://cordis.europa.eu/ist/cognition/*

This page intentionally left blank

Tutorial on Hybrid Reasoning with Argumentation Schemes

This page intentionally left blank

Knowledge-Based Software Engineering
M. Virvou and T. Nakamura (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-900-4-543

Hybrid Reasoning with Argumentation Schemes

Instructor: Dr. Thomas F. Gordon, Fraunhofer FOKUS

Abstract:

Practical reasoning typically requires a variety of argumentation schemes to be used together to solve problems and make decisions. For example, a legal case may raise issues requiring argument from precedent cases, rules, policy goals, moral principles, jurisprudential doctrine, social values and evidence. This tutorial presents an introduction to the modern philosophy of argumentation and argumentation technology, including an extensible software architecture which allows diverse computational models of argumentation schemes to be used together in an integrated way to construct and search for arguments. The architecture has been implemented in Carneades, a software library for building argumentation tools. The architecture is illustrated with models of schemes for argument from ontologies, rules, cases and testimonial evidence and compared to blackboard systems for hybrid reasoning.

This page intentionally left blank

Author Index